PB81-156267

The Ames Test Curve Fitting Program
A User's Guide

(U.S.) Health Effects Research Lab.
Research Triangle Park, NC

Sep 80

U.S. Department of Commerce
National Technical Information Service
NTIS.

United States
Environmental Protection
Agency

Health Effects Research
Laboratory
Research Triangle Park NC 27711

EPA-600/2-80-184
September 1980

Research and Development

PB81-156267

EPA

Users Guide

The Ames Test Curve Fitting Program

RESEARCH REPORTING SERIES

Research reports of the Office of Research and Development, U.S. Environmental Protection Agency, have been grouped into nine series. These nine broad categories were established to facilitate further development and application of environmental technology. Elimination of traditional grouping was consciously planned to foster technology transfer and a maximum interface in related fields. The nine series are:

1. Environmental Health Effects Research
2. Environmental Protection Technology
3. Ecological Research
4. Environmental Monitoring
5. Socioeconomic Environmental Studies
6. Scientific and Technical Assessment Reports (STAR)
7. Interagency Energy-Environment Research and Development
8. "Special" Reports
9. Miscellaneous Reports

This report has been assigned to the ENVIRONMENTAL PROTECTION TECHNOLOGY series. This series describes research performed to develop and demonstrate instrumentation, equipment, and methodology to repair or prevent environmental degradation from point and non-point sources of pollution. This work provides the new or improved technology required for the control and treatment of pollution sources to meet environmental quality standards.

EPA REVIEW NOTICE

This report has been reviewed by the U.S. Environmental Protection Agency, and approved for publication. Approval does not signify that the contents necessarily reflect the views and policy of the Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

EPA-600/2-80-184
September 1980

THE AMES TEST CURVE FITTING PROGRAM

A USER'S GUIDE

Vic Hasselblad
Andy Stead
John Creason
Violette Kasica

Biometry Division
Health Effects Research Laboratory
Environmental Protection Agency
Research Triangle Park, North Carolina 27711

DISCLAIMER

This report has been reviewed by the Health Effects Research Laboratory, U.S. Environmental Protection Agency, and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the U.S. Environmental Protection Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

Table of Contents

	Page
Abstract.	i
Acknowledgment.	ii
1. General Description	1
2. Input	2
3. Restriction	3
4. Variables in Common	3
5. Description of the Program and Subroutines.	4
6. Flow Chart of the Ames Test Program	9
7. Error Messages.	10
8. Sample Output and Explanation	11
9. References.	15
10. Listing of the Program.	16
11. Results of Simulation Runs.	45
12. Additional Examples	50

ABSTRACT

The Ames Test Curve Fitting program is a FORTRAN program which fits a nonlinear dose-response curve to count (Poisson) data. The program was written specifically for analyzing Ames test data, although it is possible that the program could have other applications. The curve fitted is of the form $[a+\exp(b+c\log X)]\exp(-dx)$ where a, b, c and d are parameters to be estimated. If the parameter d is very near zero, then d is dropped from the model, and a reduced model is fitted. Tests of goodness-of-fit and for the significance of the parameters are made using likelihood ratio tests. The program has optional graphics capability for producing plots of the data and the estimated curve.

ACKNOWLEDGMENT

The authors would like to express their appreciation for the excellent suggestions made by Larry Myers and Jerry Nehls, and for the work on the inversion routine by Michael Branton. A special thanks is due Barbara Crabtree for her patience in typing the many versions of this manual.

1. General Description

This program fits a specific nonlinear dose-response curve to count data. The program was written specifically for analyzing Ames test data, although it is possible to use the program for other applications. The program should be used by a statistician familiar with maximum likelihood techniques, including likelihood ratio tests. Some knowledge of FORTRAN is very helpful in using the program itself.

The data for the program consists of fixed dose levels and the resulting count(s) from the plates exposed to those dose levels. The observed count, y_i, measured at a dose, x_i, is assumed to follow a Poisson distribution with parameter λ_i, where λ_i is given by

$$\lambda_i = [\beta_0+\exp(\beta_1+\beta_2\log x_i)]\exp(-\beta_3 x_i) \qquad \text{for } x_i>0$$

$$\lambda_i = \beta_0 \qquad \text{for } x_i = 0$$

where x_i is the dose. The parameters β_0, β_2 and β_3 are restricted to be positive. A general description of this, and its interpretation was given by Stead, et. al. [1].

The program fits two different models. First, it estimates all four parameters β_0, β_1, β_2, β_3, and then fits a reduced model without β_3. If the likelihood ratio test for β_3 is less than .45 (the 50th percentile of the chi-square distribution with one degree of freedom) then only the reduced model is used. This is done because the β_3 term (toxicity) introduces severe computational problems if in fact there is no evidence of toxicity.

The observed counts, parameter estimates, likelihood ratio tests, and fitted curve are outputs. If a Tektronix graphics package is available, then the graph will appear on a single page with the other output. If the graphics package is not available, then the graph will appear as a printer plot on a second page.

The program was developed in the Biometry Division, HERL, of EPA by Andy Stead, Vic Hasselblad and John Creason. The program was written in ASCII FORTRAN and compiled on a UNIVAC 1110, using EXEC 8. An attempt was made to use standard FORTRAN features, so that the program could be easily converted to another machine.

2. Input

Card #	Format	Description
1	18A4	First line of title, any 72 characters
2	18A4	Second line of title, any 72 characters
3+	F10.0,1X,A1,A3,25A1	DOSE, ASK, UNIT, COUNTS: Each card contains a DOSE and up to 10 observed COUNTS. The DOSE is a floating point number. ASK is either " " or "*"; a record with an "*" is listed but ignored in the computation. (It is designed to display positive controls). UNIT is a 3 character description of the units, such as UGS, MGS, ULS. If left blank, it will be taken from the last card to have a UNIT on it, provided that card did not have an "*". The COUNTS can be anywhere in columns 16-40 with the restriction that each count must be preceded by a blank.
last	@EOF	An end of file card

Repeat the above sequence for additional data sets. The last data set should be followed by an additional @EOF card. The following is an example:

```
@XQT  VHAMTEST.ABS
SAMPLE ID: MSER790056BMGS                ACTIVATION: -
STRAIN: TA98      DATE:  8/21/79      TECHNICIAN: MJK
        0.  UGS   29   13   18
        3. *UGS  189  196  208
       60.  UGS   30   37   41
      100.  UGS   36   41   39
      200.  UGS   49   55   55
      600.  UGS   85   99   87
     1000.  UGS   90
@EOF
SAMPLE ID: CMBX790033HSHC                ACTIVATION: +
STRAIN: TA1538    DATE:  1/23/79      TECHNICIAN: LKS
        0.  UGS   16   18
       .20 *UGS  329  371
       9.9  UGS   37   55
       30.  UGS   70   79
       99.  UGS  164  174
      300.       380  360
@EOF
@EOF
```

In this example the executable program is stored in a disk file, and executed. Other ECL statements are required, but they are specific to the installation.

3. Restrictions

The input must include at least one count at the zero dose level. At least 3 distinct nonzero dose levels are needed to fit the 3 parameter model, and at least 4 distinct nonzero dose levels are needed to fit the 4 parameter model. All doses are assumed to be in the same units (except possibly those with an "*"). The doses may be in any order, and there may be more than one card with the same dose.

The number of distinct dose levels may not exceed 50. The total number of counts may not exceed 200. The program will terminate with an error message if either of these limits is exceeded. Only the first 10 counts are read from any one card.

Because of space restrictions on the graphics screen, only the first 9 dose cards will be listed. All dose cards will be used in the calculations, however.

The program requires a FORTRAN library, and if the graphics portion is used, then the Tektronix graphics package is also required.

4. Variables in Common

The main program and 4 of the subroutines use a single COMMON statement. The subroutines are AMT2, AMT3, AMT4, and PLOTIT, and these are the only variables passed between the main program and these subroutines. The variable names are identical in each case. The following gives the use of each variable:

N is the number of counts.
X is the array of doses.
Y is the array of counts
B is the vector of parameter estimates:
B(1) = β_0, B(2) = β_1, B(3) = β_2, B(4) = β_3.
V is the covariance matrix of the parameter estimates.
XLK is the likelihood of the current model.
YFACT is the sum of LOG(Y(I)!).
NIT is the number of iterations for the current model.

5. Description of the Program and Subroutines

MAIN:

The main program reads in the title and the dose levels along with the observed counts. Sums are calculated in order to test the goodness-of-fit of the Poisson assumption. The program then calls AMT3 which fits the 3 parameter model. After saving the estimates, it calls AMT4 which fits the 4 parameter model. The difference in log-likelihoods is used to decide which model to use. The quantity

$$X^2 = -2.*(L_3-L_4)$$

is asymptotically distributed as a chi-square with one degree of freedom (L_4 and L_3 are the log-likelihoods of the two respective models). If $X^2 > .454937$ (the 50th percentile) then the 4 parameter model is used. If $X^2 < .454937$, then there is no evidence of toxicity and the 3 parameter model is used.

If the 4 parameter model is fitted, then the mutagenicity must be tested using a 2 parameter model including β_0 and β_3 only. This is done by subroutine AMT2. The log-likelihoods from AMT2 and AMT4 routine are used to make the likelihood ratio test for mutagenicity.

If the 3 parameter model is fitted, then the mutagenicity is tested by comparing log-likelihood of a single mean β_0 model, calculated in the main program with that calculated in AMT3.

If any of the models fail to converge, the main program writes out

WARNING: (4, 3, or 2) PARAMETER MODEL DID NOT CONVERGE.

For an explanation of this, see the section on error messages.

Finally, the main program calls the plot subroutine, PLOTIT, if Tektronics graphics are available. If not available, then the printer plot routine, PRPLT is called.

AMT4:

This subroutine fits a non-linear curve to count data. The data are assumed to be Poisson counts with parameter λ_i, where

$$\lambda_i = [\beta_0+\exp(\beta_1+\beta_2 \log x_i)]\exp(-\beta_3 x_i) \qquad \text{for } x_i>0$$

$$\lambda_i = \beta_0 \qquad \text{for } x_i = 0$$

where x_i is the dose, and β_0, β_2 and $\beta_3>0$. The subroutine calculates the initial estimates by averaging the observed counts y_i for the zero dose. This becomes the estimate of β_0. β_0 is the then subtracted from all

the other counts, but restricted to a minimum of 5. After many simulation runs we found that the value 5 worked much better than the value 1 which may seem more logical. The logs of these adjusted counts are used to estimate β_1, β_2, and β_3 using the standard linear regression formula. If the estimate of β_3 is negative then it is set equal to 0, and just β_1 and β_2 are reestimated from the adjusted counts.

These initial estimates are used in a modified Gauss-Newton method to obtain the final estimates of the parameters. The program maximizes the log-likelihood function

$$L = \sum_{i=1}^{N} [y_i \ln(\lambda_i) - \lambda_i - \ln(y_i!)],$$

where N is the total number of plates counted. The Gauss-Newton method requires the elements of the first derivatives:

$$d_{0i} = (y_i/\lambda_i - 1)\exp(-\beta_3 x_i)$$

$$d_{1i} = (y_i/\lambda_i - 1)\exp(\beta_1 + \beta_2 \ln(x_i) - \beta_3 x_i) \qquad x_i > 0$$

$$= 0 \qquad x_i = 0$$

$$d_{2i} = \ln(x_i)(y_i/\lambda_i - 1)\exp(\beta_1 + \beta_2 \ln(x_i) - \beta_3 x_i) \qquad x_i > 0$$

$$= 0 \qquad x_i = 0$$

$$d_{3i} = -(y_i/\lambda_i - 1)x_i \lambda_i$$

Then compute the first partials:

$$D_j = \sum_{i=1}^{N} d_{ji}$$

Compute the sums of squares and cross products:

$$W_{jk} = \sum_{i=1}^{N} d_{ji} d_{ki}$$

The new estimate of $\hat{\beta}$ is given by

$$\hat{\hat{\beta}} = \hat{\beta} + W^{-1}D$$

If this change does not improve the likelihood, then the step size is cut in half until it does. If the step size gets below .00001, the program returns the last estimates. The program also returns if the number of iterations exceeds 500. If the subroutine does not iterate at all, or takes all 500 iterations, then the main program writes out the warning:

WARNING: 4 PARAMETER MODEL DID NOT CONVERGE.

AMT3:

This subroutine fits a non-linear curve to count data. The data are assumed to be Poissor counts with parameter λ_i, where

$$\lambda_i = \beta_0+\exp(\beta_1+\beta_2\log x_i) \qquad \text{for } x_i>0$$

$$\lambda_i = \beta_0 \qquad \text{for } x_i = 0$$

where x_i is the dose, and β_0 and β_2 >0. The subroutine calculates the initial estimates by averaging the observed counts y_i for the zero dose. This becomes the estimate of β_0. β_0 is the then subtracted from all the other counts, but restricted to a minimum of 5. The logs of these adjusted counts are used to estimate β_1 and β_2 using the standard linear regression formula.

These initial estimates are used in a modified Gauss-Newton method to obtain the final estimates of the parameters. The program maximizes the log-likelihood function

$$L = \sum_{i=1}^{N}[y_i \ln(\lambda_i)-\lambda_i-\ln(y_i!)],$$

where N is the total number of plates counted. The Gauss-Newton method requires the elements of the first derivatives:

$$d_{0i} = y_i/\lambda_i-1 \qquad x_i>0$$

$$d_{1i} = (y_i/\lambda_i-1)\exp(\beta_1+\beta_2\ln(x_i))$$

$$= 0 \qquad x_i = 0$$

$$d_{2i} = \ln(x_i)(y_i/\lambda_i-1)\exp(\beta_1+\beta_2\ln(x_i)) \qquad x_i>0$$

$$= 0 \qquad x_i = 0$$

Then compute the first partials:

$$D_j = \sum_{i=1}^{N} d_{ji}$$

Compute the sums of squares and cross products:

$$V_{jk} = \sum_{i=1}^{N} d_{ji}d_{ki}$$

The new estimate of $\hat{\beta}$ is given by

$$\hat{\hat{\beta}} = \hat{\beta} + V^{-1}D$$

If this change does not improve the likelihood, then the step size is cut in half until it does. If the step size gets below .00001, the program returns the last estimates. The program also returns if the number of iterations exceeds 500. If the subroutine does not iterate at all, or takes all 500 iterations, then the main program writes out the warning:

WARNING: 3 PARAMETER MODEL DID NOT CONVERGE.

AMT2:

This subroutine fits a non-linear curve to count data. The data are assumed to be Poisson counts with parameter λ_i, where

$$\lambda_i = \exp(\beta_0 - \beta_3 x_i)$$

where x_i is the dose, and β_0 and $\beta_3 > 0$. The subroutine calculates the initial estimates using the standard linear regression formula on the logs of the counts. If the estimate of β_3 is negative then it is set equal to 0, and just β_0 is reestimated from the adjusted counts. These initial estimates are used in a Newton-Raphson method to obtain the final estimates of the parameter. The Newton-Raphson method requires the first and second partial derivatives. Let

$$D_0 = \sum_{i=1}^{N} y_i - \lambda_i$$

$$D_1 = \sum_{i=1}^{N} -x_i(y_i - \lambda_i)$$

$$V_{00} = \sum_{i=1}^{N} \lambda_i$$

$$V_{01} = V_{10} = \sum_{i=1}^{N} \lambda_i x_i$$

$$V_{11} = \sum_{i=1}^{N} \lambda_i x_i^2$$

Then

$$\hat{\hat{\beta}} = \hat{\beta} + V^{-1}D$$

This method is continued until the combined relative change in β_0 and β_3 is less than .0001. The program returns if the number of iterations exceeds 200. If the subroutine does not iterate at all, or takes all 200 iterations, then the main program writes out the warning:

WARNING: 2 PARAMETER MODEL DID NOT CONVERGE.

INVSPD(A,B,N,DET):

This is a matrix inversion routine, along with the accompanying solution for linear equations. The program returns A^{-1} in A, and $A^{-1}B$ in B. N is the dimension of B (and therefore A is N x N). The routine uses a Gauss-Jordan reduction algorithm which computes the inverse in place.

CHISQ(X,N):

This function computes the right hand tail probability of the chi-square function with N degrees of freedom. For N even, the formula is exact. For N odd, the routine needs the cumulative normal function (CNORM). The formulas are given by Abramowitz and Stegun [3].

CNORM(X):

This function computes the cumulative normal distribution from minus infinity to X. The routine uses a rational fraction approximation, which is given by Hastings [4].

ALGMA(X):

This function computes the natural logarithm of the gamma function. It uses Stirling's approximation which is given by Abramowitz and Stegun [3].

INTRP(NCD,NZ,K):

This subroutine receives 25 alphabetic characters in the array NCD. The array is scanned for non-negative integers, each one of which must be preceeded by a blank. The first ten integers are returned in NZ. The number of integers found is returned in K.

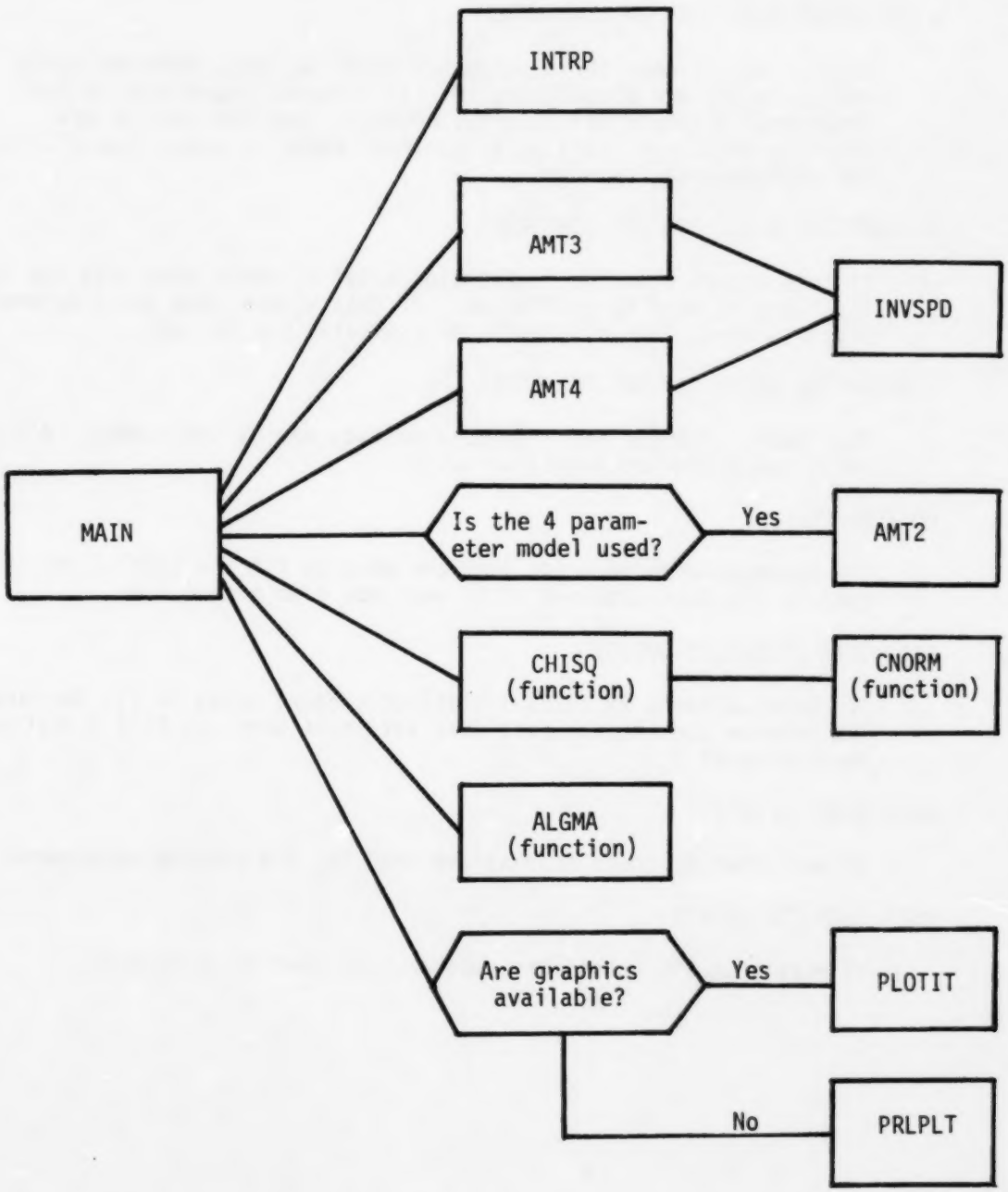
PLOTIT:

This subroutine creates the plot of the data points, as well as the fitted curve. The curve is evaluated at 100 points.

PRPLT(NUNP):

This subroutine prints a printer plot of the observed counts and fitted curve. NUNP is the label used on the horizontal axis.

6. Flow Chart of the Ames Test Program



7. Error Messages

UNEXPECTED END OF FILE ENCOUNTERED:

This occurs if an end of file is found when reading either the card with the maximum number of counts (2nd card) or the variable format card (3rd card). The program terminates if this happens.

4 PARAMETER MODEL DID NOT CONVERGE:

If this occurs when the 3 parameter model is used, then the tests and estimates are probably valid. If a visual inspection of the data shows evidence of toxicity, however, then the results are probably very poor. If the 4 parameter model is used, then all tests and estimates may be poor.

3 PARAMETER MODEL DID NOT CONVERGE:

If this occurs when the 4 parameter model is used, then only the test for toxicity need be questioned. If this occurs when the 3 parameter model is used, then all tests and estimates may be poor.

2 PARAMETER MODEL DID NOT CONVERGE:

This means that the test for mutagenicity may be inaccurate. All other tests and estimates should be valid.

NO ZERO DOSE:

The program needs at least one zero dose to fit the model. The program goes to the next data set if it does not find a zero dose.

LESS THAN 3 NONZERO DOSES:

The program needs at least 3 distinct nonzero doses to fit the model. The program goes to the next data set if it does not find 3 distinct nonzero doses.

MORE THAN 50 DOSES:

If more than 50 doses (cards) are read in, the program terminates.

MORE THAN 200 COUNTS:

If more than 200 counts are read in, the program terminates.

(1) SAMPLE ID: MSER790056BMGS ACTIVATION: -
STRAIN: TA98 DATE: 8/21/79 TECHNICIAN: MJK

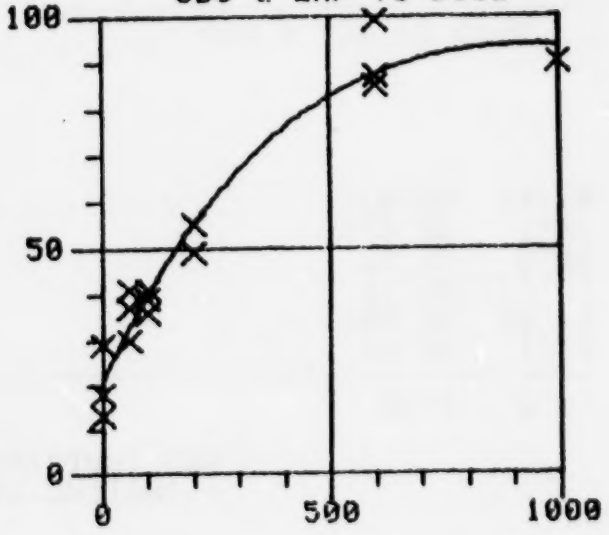
(2) DOSE	(3) UNITS	(4) PLATE COUNTS			(5) MEAN	(6) S.D.
.00	UGS	29	13	18	20.00	8.19
3.00*	UGS	189	196	208	197.67	9.61
60.00	UGS	30	37	41	36.00	5.57
100.00	UGS	36	41	39	38.67	2.52
200.00	UGS	49	55	55	53.00	3.46
600.00	UGS	85	99	87	90.33	7.57
1000.00	UGS	90			90.00	.00

	B(0)	B(1)	B(2)	B(3)
(7) ESTS.	20.489	-1.2629	.9618	.00093

TEST	CHI-SQUARE	DF	P	LOGL
(8) POISSON	10.47	10	.4001	-50.1233
(9) ADEQUACY	1.65	2	.4377	-50.9496
(10) TOXICITY	3.24	1	.0717	-52.5718
(11) MUTAGENICITY	44.36	2	.0000	-73.1293

(12) AVERAGE SLOPE (NONLIN. MODEL) = .221
95% CONF. LIMITS = (.051, .957)

(13) AVERAGE SLOPE (LINEAR REGR.) = .110
95% CONF. LIMITS = (.094, .126)

(14)


Second example resulting from the input shown in section 2

SAMPLE ID: CMBX790033HSHC ACTIVATION: +
STRAIN: TA1538 DATE: 1/23/79 TECHNICIAN: LKS

DOSE	UNITS	PLATE COUNTS		MEAN	S.D.
.00	UGS	16	18	17.00	1.41
.20*	UGS	329	371	350.00	29.70
9.90	UGS	37	55	46.00	12.73
30.00	UGS	70	79	74.50	6.36
99.00	UGS	164	174	169.00	7.07
300.00	UGS	380	360	370.00	14.14

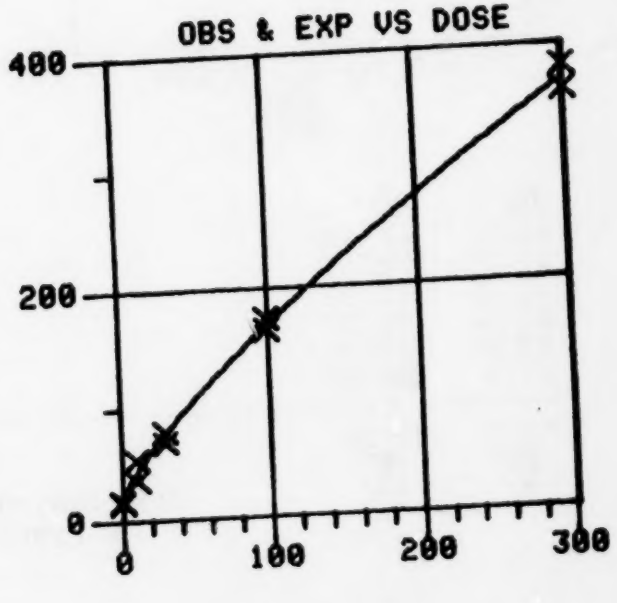
	B(0)	B(1)	B(2)
(15) ESTS.	17.396	1.5033	.7645

NO EVIDENCE OF TOXICITY

TEST	CHI-SQUARE	DF	P	LOGL
POISSON	5.02	5	.4135	-33.7202
ADEQUACY	.66	2	.7204	-34.0480
MUTAGENICITY	1121.21	2	.0000	-594.6514

AVERAGE SLOPE (NONLIN. MODEL) = 1.174
95% CONF. LIMITS = (1.017, 1.355)

AVERAGE SLOPE (LINEAR REGR.) = 1.141
95% CONF. LIMITS = (1.048, 1.234)



Explanation of numbered items:

(1) The 2 card-72 character title read in.

(2) The dose levels (X(I))

(3) The units: UGS = micrograms, MGS = milligrams, ULS = microliters, NGS = nanograms

(4) The observed counts as read in

(5) The mean of the counts for the particular dose

(6) The standard deviation of the counts.

(7) The maximum likelihood parameter estimates. If the full model is used, 4 parameter estimates will be given. Otherwise, 3 parameter estimates will be given.

(8) The chi-square for the Poisson assumption, and its associated p-value, give an estimate of the validity of the Poisson assumption. If the p-value is very small (<.05, say) then there is evidence of too much variation for the assumption to hold. The parameter estimates and the other tests may still be valid, but they should be viewed with caution. Also, the confidence limits in (12) may not be wide enough.

(9) This chi-square tests the validity of the form of the curve. It is the result of a likelihood ratio test using the likelihood of the all means model vs. the likelihood of the fitted model. The degrees of freedom will be the number of dose levels (including 0) minus the number of parameters fitted (6 - 4 = 2) in this case. If there are only 3 non-zero dose levels, this statistic will not appear. The chi-square may be non-significant even if the Poisson assumption is not met. This would imply that curve is a good description of the mean counts for each dose. If the chi-square in (9) is significant, then the curve is not a good one, and tests in (10) and (11) should be viewed with caution. The average slope may not be badly biased, but the confidence limits probably are not accurate.

(10) This chi-square tests the signficance of the toxicity term (β_3). The test is computed from the differences in the likelihoods of the 4 parameter and 3 parameter models. This test only appears if the 4 parameter model was used. The presence of toxicity greatly increases the variances of the other parameter estimates, and therefore increases the width of the confidence interval for the average slope.

(11) This chi-square tests the significance of mutagenicity terms (β_1, β_2). They are tested jointly using a likelihood ratio test. If a 4 parameter model is used, then the test is the difference of the likelihoods of the 4 parameter model and the 2 parameter model (β_0 and β_3 only).

If a 3 parameter model is used, then the test is the difference of the likelihoods of the 3 parameter model and a common mean model (β_0 only).

(12) The average slope (nonlinear model) is computed as follows:

$$A.S. = [\exp(\beta_1+\beta_2 Xm)]/Xm.$$

It represents the increase over background at the maximum dose Xm, divided by the dose Xm. In other words, it represents the average slope of the curve from 0 to the point Xm. The 95 percent confidence interval comes from the estimated asymptotic variances and covariance of β_1 and β_2.

(13) The average slope (linear regression) is computed directly from the counts. All points within a single dose are discarded if the average count is more than one standard deviation below the previous average count. At least 3 dose levels are retained. The average slope is the slope of a simple linear regression run on the non-discarded data. The 95% confidence interval is calculated from the standard error of the estimate of the slope.

(14) If the graphics package is available the program plots the observed counts and the fitted curve.

(15) The second example has no evidence of toxicity, and so the program only estimates 3 parameters.

9. References

[1] Stead, A. G., et al, "Modeling the Ames Test", to appear in Mutagenic Research.

[2] Hartley, H. O., "The Modified Gauss-Newton Method for the Fitting of Non-Linear Regression Functions by Least Squares", Technometrics, 3, 269-280, 1961.

[3] Abramowitz, M., and I. A. Stegun, Handbook of Mathematical Functions, Dover Publications, 1965.

[4] Hastings, C., Approximations for Digital Computers, Princeton University Press, p. 169, 1955.

10. Listing of the Program

```
C     AMESTEST CURVE FITTING PROGRAM                    REVISED  8-27-80
C        PROGRAMMERS: STEAD, HASSELBLAD, CREASON
C     FITS A POISSON DISTRIBUTION WITH PARAMETER L(I), WHERE
C
C        L(I) = (B(1)+EXP(B(2)+B(3)LOG(X(I))EXP(-B(4)X(I))
C
      DIMENSION TITLE(36), NCD(25), Z(10), NZ(10), BP(4), VP(4,4)
      DIMENSION CKS(50), XMS(50), SSS(50), SDS(50), XDS(50)
      COMMON N, X(200), Y(200), B(4), V(4,4), XLK, YFACT, NIT
C
C     N IS THE NUMBER OF COUNTS
C     X IS THE ARRAY OF DOSES
C     Y IS THE ARRAY OF COUNTS
C     B IS THE VECTOR OF PARAMETER ESTIMATES:
C        B(1) = BETA(0), B(2) = BETA(1), B(3) = BETA(2), B(4) = BETA(3)
C     V IS THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATES
C     XLK IS THE LIKELIHOOD OF THE CURRENT MODEL
C     YFACT IS THE SUM OF LOG(Y(I)!)
C     NIT IS THE NUMBER OF ITERATIONS FOR THE CURRENT MODEL
C
C     CHIS AND CHSQ ARE CHI-SQUARES FOR VARIOUS TESTS
C     BP AND VP SAVE THE VALUES OF B AND V FROM A PREVIOUS MODEL
C     XM AND XM2 ARE USED TO COMPUTE A GOODNESS OF FIT STATISTIC
C     XLKM IS THE LIKELIHOOD FOR THE ALL MEANS MODEL
C     XLKF IS THE LIKELIHOOD FOR THE 4 PARAMETER MODEL
C     XLKR IS THE LIKELIHOOD FOR THE 3 PARAMETER MODEL
C     XLK2 IS THE LIKELIHOOD FOR THE 2 PARAMETER MODEL
C     NUNT IS THE ALPHABETIC UNIT CODE READ IN
C     NUNP IS THE LAST NONBLANK ALPHABETIC UNIT CODE READ IN
C     NCARD IS THE NUMBER OF CARDS READ IN
C
C     THE FOLLOWING ARE USED IN THE SPECIFIC ACTIVITY CALCULATION
C     NDOSE IS THE TOTAL NUMBER OF DOSE LEVELS
C     NCTL IS THE NUMBER OF DISTINCT NONZERO DOSE LEVELS
C     CKS IS THE ARRAY OF K'S FOR EACH DOSE
C     XMS IS THE ARRAY OF MEAN COUNTS FOR EACH DOSE
C     SSS IS THE ARRAY OF SUMS OF SQUARES FOR EACH DOSE
C     SDS IS THE ARRAY OF STANDARD DEVIATIONS FOR EACH DOSE
C     XDS IS THE ARRAY OF DOSES
C
C     IPLOT IS A VARIABLE FIXED AT COMPILE TIME TO INDICATE WHETHER
C        TEKTRONIX GRAPHICS ARE AVAILABLE (IPLOT = 1), OR NOT
C        AVAILABLE (IPLOT = 0)
C
      DATA NASK,NBL/'*',' '/
      DATA IPLOT/0/
C
C     SET IPLOT = 0 IF GRAPHICS TERMINAL NOT AVAILABLE
C
 2    CONTINUE
```

```
C
C     INITIALIZE PLOT PARAMETERS
C
      IF(IPLOT.EQ.1) CALL INITT(120)
      IF(IPLOT.EQ.1) CALL BINITT
      READ (5,5,END=900) TITLE
 5    FORMAT(18A4)
      WRITE (6,10) TITLE
 10   FORMAT('1',18A4/' ',18A4)
      NUNP = NBL
      YFACT = 0.
      XLKM = 0.
      CHIS = 0.
      YSUM = 0.
      NDF = 0
      WRITE (6,25)
 25   FORMAT('0',14X,'DOSE UNITS  PLATE COUNTS',23X,'MEAN   S.D.')
      WRITE (6,30)
 30   FORMAT(' ',11X,61('-'))
      I = 0
      N = 0
      NCARD = 0
C
C     READ IN DOSE, UNITS, COUNTS
C
 3    CONTINUE
      READ (5,15,END=4) XD, NSK, NUNT, NCD
 15   FORMAT(F10.0,1X,A1,A3,25A1)
      NCARD = NCARD + 1
      IF(NCARD.EQ.10) WRITE (6,120)
 120  FORMAT(' MORE THAN 9 DOSE LEVELS USED IN COMPUTATION BUT NOT DISPL
     1AYED')
      IF(NUNT.EQ.NBL) NUNT = NUNP
      IF(NUNT.NE.NBL.AND.NSK.NE.NASK) NUNP = NUNT
C
C     CONVERT ALPHABETICS TO INTEGERS
C
      CALL INTRP(NCD,NZ,K)
      XM = 0.
      XM2 = 0.
      CK = K
C
C     CHECK FOR CASES NOT INCLUDED
C
      IF(K.GT.0.AND.NSK.NE.NASK) GO TO 6
      IF(K.EQ.0.AND.NCARD.LE.9) WRITE (6,20) XD, NSK, NUNT, NCD
      IF(K.EQ.0) GO TO 3
      DO 7 J = 1, K
         XM = XM + NZ(J)
         XM2 = XM2 + NZ(J)**2
```

```
 7    CONTINUE
      STD = 0.
      XM = XM/CK
      IF(K.GT.1) STD = SQRT((XM2 - XM**2*CK)/(CK - 1.))
      IF(NCARD.LE.9) WRITE (6,20) XD, NSK, NUNT, NCD, XM, STD
      GO TO 3
C
C     CREATE ARRAYS OF DOSES AND COUNTS WITH ONE DOSE FOR EACH COUNT
C     COMPUTE SUMS, SQUARES FOR TESTING THE POISSON ASSUMPTION
C     COMPUTE SUM(LOG(Y(I)!)) FOR LOG-LIKELIHOOD FUNCTIONS
C
 6    CONTINUE
      I = I + 1
      IF(I.GT.50) GO TO 940
      DO 8 J = 1, K
         N = N + 1
         IF(N.GT.200) GO TO 950
         Z(J) = NZ(J)
         X(N) = XD
         Y(N) = Z(J)
         XM = XM + Z(J)
         XM2 = XM2 + Z(J)**2
         YFACT = YFACT + ALGMA(Z(J) + 1.)
         YSUM = YSUM + Z(J)
 8    CONTINUE
      XM = XM/CK
      STD = 0.
      IF(K.GT.1) STD = SQRT((XM2 - XM**2*CK)/(CK - 1.))
      CKS(I) = CK
      XMS(I) = XM
      SSS(I) = XM2
      SDS(I) = STD
      XDS(I) = XD
C
C     COMPUTE CHI-SQUARE FOR TESTING THE POISSON ASSUMPTION
C
      CHIS = CHIS + XM2/XM - XM*CK
      NDF = NDF + K - 1
C
C     COMPUTE LOG-LIKELIHOOD FOR ALL MEANS MODEL
C
      DO 9 J = 1, K
         XLKM = XLKM - XM + Z(J)*ALOG(XM)
 9    CONTINUE
      IF(NCARD.LE.9) WRITE (6,20) XD, NSK, NUNT, NCD, XM, STD
 20   FORMAT(11X,F8.2,A1,1X,A3,25A1,5X,F12.2,F7.2)
      GO TO 3
C
C     END OF DATA READING SECTION
C
```

```
 4    CONTINUE
      NDOSE = I
      IF(NDOSE.LE.1) GO TO 930
      IF(CHIS.LE.0.) CHIS = .01
C
C     SPACE DOWN
C
      IF(NCARD.GT.9) GO TO 22
      K = 10 - NCARD
      DO 23 J = 1, K
         WRITE (6,35)
 35      FORMAT(' ')
 23   CONTINUE
 22   CONTINUE
C
C     SORT DOSES FOR SPECIFIC ACTIVITY COMPUTATION
C
      NCTL = 0
      DO 13 I = 1, NDOSE
         DO 14 J = I, NDOSE
            IF(XDS(I).LE.XDS(J)) GO TO 14
            TEMP = XDS(I)
            XDS(I) = XDS(J)
            XDS(J) = TEMP
            TEMP = CKS(I)
            CKS(I) = CKS(J)
            CKS(J) = TEMP
            TEMP = XMS(I)
            XMS(I) = XMS(J)
            XMS(J) = TEMP
            TEMP = SSS(I)
            SSS(I) = SSS(J)
            SSS(J) = TEMP
            TEMP = SDS(I)
            SDS(I) = SDS(J)
            SDS(J) = TEMP
 14       CONTINUE
          IF(I.EQ.1) GO TO 13
          IF(XDS(I).NE.XDS(I-1)) NCTL = NCTL + 1
 13   CONTINUE
      IF(NCTL.LT.3) GO TO 930
      IF(XDS(1).NE.0.) GO TO 920
C
C     FIND HIGHEST DOSE WITH NO EVIDENCE OF TOXICITY
C
      MDOSE = NDOSE
      K = NDOSE - 3
      IF(K.LE.0) GO TO 16
      DO 17 J = 1, K
         I = NDOSE + 1 - J
```

```
         IF(XMS(I).GT.XMS(I-1)+SDS(I-1)) GO TO 18
 17    CONTINUE
 18    CONTINUE
       MDOSE = I
 16    CONTINUE
C
C      COMPUTE AVERAGE SLOPE UP TO X(MDOSE)
C
       CKSM = 0.
       XM = 0.
       XM2 = 0.
       YS = 0.
       Y2S = 0.
       XYS = 0.
       DO 19 J = 1, MDOSE
          CKSM = CKSM + CKS(J)
          YS = YS + XMS(J)*CKS(J)
          Y2S = Y2S + SSS(J)
          XM = XM + XDS(J)*CKS(J)
          XM2 = XM2 + XDS(J)**2*CKS(J)
          XYS = XYS + XDS(J)*XMS(J)*CKS(J)
 19    CONTINUE
       VX = XM2 - XM**2/CKSM
       Y2S = Y2S - YS**2/CKSM
       SPCL = (XYS - XM*YS/CKSM)/VX
       VSPL = (Y2S/VX - SPCL**2)/(CKSM - 2.)
       XLKM = XLKM - YFACT
       PVAL = CHISQ(CHIS,NDF)
C
C      FIT THE REDUCED (3 PARAMETER) MODEL
C
       CALL AMT3
       NIT3 = NIT
C
C      SAVE THE ESTIMATES
C
       DO 11 J = 1, 4
          BP(J) = B(J)
          DO 12 K = 1, 4
             VP(J,K) = V(J,K)
 12       CONTINUE
 11    CONTINUE
       XLKR = XLK
       IF(NCTL.EQ.3) GO TO 31
C
C      FIT THE FULL (4 PARAMETER) MODEL
C
       CALL AMT4
       XLKF = XLK
       NIT4 = NIT
```

```
C
C     COMPUTE LIKELIHOOD RATIO TEST FOR THE DIFFERENCE IN THE MODELS
C
      CHSQ = 2.*(XLKF - XLKR)
      IF(CHSQ.LT.0.454937) GO TO 31
C
C     USE 4 PARAMETER MODEL
C
      WRITE (6,40)
 40   FORMAT(' ',10X,'B(0)',5X,'B(1)',4X,'B(2)',5X,'B(3)'/
     1 11X,'----',5X,'----',4X,'----',5X,'----')
      DO 32 J = 1, 4
         BP(J) = B(J)
 32   CONTINUE
      WRITE (6,50) B
 50   FORMAT(' ESTS.',F9.3,F9.4,F8.4,F9.5)
      WRITE (6,55)
 55   FORMAT('OTEST',6X,'CHI-SQUARE',1X,'DF',5X,'P',7X,'LOGL'/
     1 ' ----',6X,'----------',1X,'--',5X,'-',7X,'----')
      IF(NDF.GT.0) WRITE (6,90) CHIS, NDF, PVAL, XLKM
 90   FORMAT(' POISSON',5X,F8.2,I3,F6.4,F11.4)
C
C     TEST FOR ADEQUACY OF THE MODEL
C
      NDF = NDOSE - 4
      CHIS = 2.*(XLKM - XLKF)
      IF(CHIS.LE.0.) CHIS = .01
      PVAL = CHISQ(CHIS,NDF)
      IF(NDF .GT. 0) WRITE (6,95) CHIS, NDF, PVAL, XLKF
 95   FORMAT(' ADEQUACY',4X,F8.2,I3,F6.4,F11.4)
      ND = 1
      PVAL = CHISQ(CHSQ,ND)
      WRITE (6,105) CHSQ, ND, PVAL, XLKR
 105  FORMAT(' TOXICITY',4X,F8.2,I3,F6.4,F11.4)
C
C     COMPUTE LIKELIHOOD FOR MODEL WITHOUT B(1),B(2) (2 PARAMETER MODEL)
C
      CALL AMT2
      XLK2 = XLK
      NIT2 = NIT
      DO 21 J = 1, 4
         B(J) = BP(J)
 21   CONTINUE
C
C     TEST FOR MUTAGENICITY
C
      CHSQ = 2.*(XLKF - XLK2)
      IF(CHSQ.LE.0.) CHSQ = .01
      ND = 2
      PVAL = CHISQ(CHSQ,ND)
```

```
      WRITE (6,110) CHSQ, ND, PVAL, XLK2
  110 FORMAT(' MUTAGENICITY',F8.2,I3,F6.4,F11.4)
      SPCA = XDS(MDOSE)
C
C     COMPUTE AVERAGE SLOPE
C
      TEMP = BP(2) + BP(3)*ALOG(SPCA)
      SPC = EXP(TEMP)/SPCA
      VSC = V(2,2) + V(3,3)*(ALOG(SPCA))**2 + 2.*V(2,3)*ALOG(SPCA)
      S = 2.*SQRT(ABS(VSC))
      IF(S.GT.50.) S = 50.
      UCL = EXP(TEMP + S)/SPCA
      ACL = EXP(TEMP - S)/SPCA
      WRITE (6,65) SPC
   65 FORMAT('0AVERAGE SLOPE (NONLIN. MODEL) =',F9.3)
      WRITE (6,70) ACL, UCL
   70 FORMAT(3X,'95% CONF. LIMITS = (',F8.3,',',F8.3,')')
      S = 2.*SQRT(VSPL)
      UCL = SPCL + S
      ACL = SPCL - S
      WRITE (6,115) SPCL
  115 FORMAT('0AVERAGE SLOPE (LINEAR REGR.) =',F10.3)
      WRITE (6,70) ACL, UCL
      GO TO 200
   31 CONTINUE
C
C     USE 3 PARAMETER MODEL
C
      WRITE (6,45)
   45 FORMAT('0',10X,'B(0)',5X,'B(1)',4X,'B(2)'/
     1 11X,'----',5X,'----',4X,'----')
      B(4) = 0.
      DO 33 J = 1, 3
         B(J) = BP(J)
   33 CONTINUE
      WRITE (6,50) (B(J),J=1,3)
      IF(NCTL.GT.3) WRITE (6,85)
   85 FORMAT('ONO EVIDENCE OF TOXICITY')
      WRITE (6,55)
C
C     COMPUTE LIKELIHOOD FOR MODEL WITH B(0) ONLY
C
      XLK = -YFACT
      XLM = YSUM/N
      DO 34 I = 1, N
         XLK = XLK - XLM + Y(I)*ALOG(XLM)
   34 CONTINUE
      IF(NDF.GT.0) WRITE (6,90) CHIS, NDF, PVAL, XLKM
C
C     TEST FOR ADEQUACY OF THE MODEL
```

```
C
      NDF = NDOSE - 3
      CHIS = 2.*(XLKM - XLKR)
      IF(CHIS.LE.0.) CHSQ = 0.01
      PVAL = CHISQ(CHIS,NDF)
      WRITE (6,95) CHIS, NDF, PVAL, XLKR
C
C     TEST FOR MUTAGENICITY
C
      XLK2 = XLK
      CHSQ = 2.*(XLKR - XLK2)
      IF(CHSQ.LE.0.) CHSQ = 0.01
      ND = 2
      PVAL = CHISQ(CHSQ,ND)
      WRITE (6,110) CHSQ, ND, PVAL, XLK2
      SPCA = XDS(MDOSE)
C
C     COMPUTE AVERAGE SLOPE
C
      TEMP = BP(2) + BP(3)*ALOG(SPCA)
      SPC = EXP(TEMP)/SPCA
      VSPC = VP(2,2) + VP(3,3)*(ALOG(SPCA))**2 + 2.*VP(2,3)*ALOG(SPCA)
      S = 2.*SQRT(ABS(VSPC))
      IF(S.GT.50.) S = 50.
      UCL = EXP(TEMP + S)/SPCA
      ACL = EXP(TEMP - S)/SPCA
      WRITE (6,65) SPC
      WRITE (6,70) ACL, UCL
      S = 2.*SQRT(VSPL)
      UCL = SPCL + S
      ACL = SPCL - S
      WRITE (6,115) SPCL
      WRITE (6,70) ACL, UCL
  200 CONTINUE
      NP = 4
      IF(NIT4.EQ.1.OR.NIT4.EQ.500.OR.VSC.LE.0.) WRITE (6,205) NP
  205 FORMAT(' WARNING:',I2,' PARAMETER MODEL DID NOT CONVERGE')
      NP = 3
      IF(NIT3.EQ.1.OR.NIT3.EQ.500.OR.VSPC.LE.0.) WRITE (6,205) NP
      NP = 2
      IF(NIT2.EQ.1.OR.NIT2.EQ.200) WRITE (6,205) NP
C
C     CALL GRAPHICS SUBROUTINE
C
      IF(IPLOT.EQ.1) CALL PLOTIT
      IF(IPLOT.EQ.1) GO TO 2
      WRITE (6,10) TITLE
      CALL PRPLT(NUNP)
      GO TO 2
  900 CONTINUE
```

```
      WRITE (6,210)
  210 FORMAT('1')
      STOP
  910 CONTINUE
      WRITE (6,905)
  905 FORMAT('0UNEXPECTED END OF FILE ENCOUNTERED')
      STOP
  920 CONTINUE
      WRITE (6,915)
  915 FORMAT('0NO ZERO DOSE')
      GO TO 2
  930 CONTINUE
      WRITE (6,925)
  925 FORMAT('0LESS THAN 3 NONZERO DOSES')
      GO TO 2
  940 CONTINUE
      WRITE (6,935)
  935 FORMAT('0MORE THAN 50 DOSES')
      STOP
  950 CONTINUE
      WRITE (6,945)
  945 FORMAT('0MORE THAN 200 COUNTS')
      STOP
      END
```

```
C     AMESTEST CURVE FIT - 4 PARAMETER MODEL          REVISED  6-24-80
C        PROGRAMMERS: HASSELBLAD, STEAD, CREASON
      SUBROUTINE AMT4
      DIMENSION XTY(4), XTX(4,4), D(4), BP(4), XL(50), XTXP(4,4),XTYP(4)
      COMMON N, X(200), Y(200), B(4), V(4,4), XLK, YFACT, NIT
C
C     N IS THE NUMBER OF COUNTS
C     X IS THE ARRAY OF DOSES
C     Y IS THE ARRAY OF COUNTS
C     B IS THE VECTOR OF PARAMETER ESTIMATES:
C        B(1) = BETA(0), B(2) = BETA(1), B(3) = BETA(2), B(4) = BETA(3)
C     V IS THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATES
C     XLK IS THE LIKELIHOOD OF THE CURRENT MODEL
C     YFACT IS THE SUM OF LOG(Y(I)!)
C     NIT IS THE NUMBER OF ITERATIONS FOR THE CURRENT MODEL
C
C     D IS THE VECTOR OF TERMS FOR THE 1ST PARTIALS AND THEIR SQUARES
C     XTX IS THE VECTOR OF 1ST PARTIALS
C     XTX IS THE MATRIX OF CROSS PRODUCTS OF 1ST PARTIALS
C     DELTA IS THE SCALAR WHICH DETERMINES THE LENGTH OF MOVEMENT
C     DOPT IS THE OPTIMAL LENGTH OF MOVEMENT
C
      NZ = 0
      BP(1) = 0.
      B(1) = 0.
C
C     COMPUTE INITIAL ESTIMATES
C
      DO 2 I = 1, N
         IF(X(I).GT.0.) GO TO 2
         NZ = NZ + 1
         B(1) = B(1) + Y(I)
 2    CONTINUE
      B(1) = B(1)/FLOAT(NZ)
      DO 3 J = 1, 3
         XTY(J) = 0.
         DO 4 K = 1, 3
            XTX(J,K) = 0.
 4       CONTINUE
 3    CONTINUE
      DO 6 I = 1, N
      IF(X(I).LE.0.) GO TO 6
         D(1) = 1.
         D(2) = ALOG(X(I))
         D(3) = -X(I)
         YP = ALOG(AMAX1(5.,Y(I)-B(1)))
         DO 7 J = 1, 3
            XTY(J) = XTY(J) + D(J)*YP
            DO 8 K = 1, 3
               XTX(J,K) = XTX(J,K) + D(J)*D(K)
```

```
 8          CONTINUE
 7        CONTINUE
 6      CONTINUE
C
C      SAVE MATRICES IN CASE INITIAL ESTIMATES NEED BE RECOMPUTED
C
       XTXP(1,1) = XTX(1,1)
       XTXP(1,2) = XTX(1,2)
       XTXP(2,1) = XTX(2,1)
       XTXP(2,2) = XTX(2,2)
       XTYP(1) = XTY(1)
       XTYP(2) = XTY(2)
       CALL INVSPD(XTX,XTY,3,DET)
C
C      CONSTRAIN INITIAL ESTIMATES TO BE POSITIVE
C
       B(2) = XTY(1)
       B(3) = AMAX1(0.,XTY(2))
       B(4) = AMAX1(0.,XTY(3))
       IF(B(4).GT.0.) GO TO 26
C
C      RECOMPUTE INITIAL ESTIMATES
C
       CALL INVSPD(XTXP,XTYP,2,DET)
       B(2) = XTYP(1)
       B(3) = AMAX1(0.,XTYP(2))
 26    CONTINUE
C
C      ITERATE USING A GAUSS-NEWTON METHOD TO FIND THE MAXIMUM
C      LIKELIHOOD SOLUTION
C
       DO 20 IT = 1, 500
       NIT = IT
       XLK1 = -YFACT
       DELTA = 1.
C
C      COMPUTE PREDICTED LAMBDAS, CURRENT LIKELIHOOD
C
       DO 9 I = 1, N
          XL(I) = B(1)
          IF(X(I).GT.0.) XL(I) = (B(1)+EXP(B(2)+B(3)*ALOG(X(I))))
      1    *EXP(-B(4)*X(I))
          IF(XL(I).EQ. 0.) XL(I) = 1.E-18
          XLK1 = XLK1 + Y(I)*ALOG(XL(I)) - XL(I)
 9     CONTINUE
       DO 11 J = 1, 4
          XTY(J) = 0.
          DO 12 K = 1, 4
             XTX(J,K) = 0.
 12       CONTINUE
```

```
 11   CONTINUE
C
C     COMPUTE ELEMENTS OF FIRST PARTIAL DERIVATIVES
C     SUM THESE AND THEIR CROSS PRODUCTS
C
      DO 14 I = 1, N
         TERM = Y(I)/XL(I) - 1.
         D(1) = TERM*EXP(-B(4)*X(I))
         D(2) = 0.
         IF(X(I).GT.0.) D(2)=TERM*EXP(B(2)+B(3)*ALOG(X(I))-B(4)*X(I))
         D(3) = 0.
         IF(X(I).GT.0.) D(3) = ALOG(X(I))*EXP(B(2)+B(3)*ALOG(X(I))-
     1    B(4)*X(I))*TERM
         D(4) = -TERM*X(I)*XL(I)
         DO 15 J = 1, 4
            XTY(J) = XTY(J) + D(J)
            DO 16 K = 1, 4
               XTX(J,K) = XTX(J,K) + D(J)*D(K)
 16         CONTINUE
 15      CONTINUE
 14   CONTINUE
C
C     COMPUTE CHANGE IN ESTIMATES
C
      CALL INVSPD(XTX,XTY,4,DET)
C
C     COMPUTE CONSTRAINED NEW ESTIMATES, AND TRY THEM
C
      DO 17 J = 1, 4
         BP(J) = B(J) + XTY(J)
         IF(J.NE.2) BP(J)=AMAX1(BP(J),0.)
 17   CONTINUE
      XLK2 = -YFACT
      DO 18 I = 1, N
         XL(I) = BP(1)
         IF(X(I).LE.0.) GO TO 28
         TEMP = BP(2) + BP(3)*ALOG(X(I))
         IF(TEMP.GT.50.) TEMP = 50.
         XL(I) = (BP(1) + EXP(TEMP))*EXP(-BP(4)*X(I))
 28      CONTINUE
         IF(XL(I).EQ.0.) XL(I) = 1.E-18
         XLK2 = XLK2 + Y(I)*ALOG(XL(I)) - XL(I)
 18   CONTINUE
C
C     FIND BEST ESTIMATES USING A CHOPPING METHOD
C
 19   CONTINUE
      XLK3 = XLK2
      DELTA = DELTA*0.5
C
```

```
C     IF CHOPPING GETS TOO SMALL, QUIT
C
      IF(DELTA.LE.0.00001) GO TO 23
      DO 21 J = 1, 4
         BP(J) = B(J) + DELTA*XTY(J)
         IF(J.NE.2) BP(J)=AMAX1(BP(J),0.)
 21   CONTINUE
      XLK2 = -YFACT
      DO 22 I = 1, N
         XL(I) = BP(1)
         IF(X(I).LE.0.) GO TO 29
         TEMP = BP(2) + BP(3)*ALOG(X(I))
         IF(TEMP.GT.50.) TEMP = 50.
         XL(I) = (BP(1) + EXP(TEMP))*EXP(-BP(4)*X(I))
 29      CONTINUE
         IF(XL(I).EQ.0.) XL(I) = 1.E-18
         XLK2 = XLK2 + Y(I)*ALOG(XL(I)) - XL(I)
 22   CONTINUE
      IF(XLK2.LE.XLK1) GO TO 19
C
C     OPTIMIZE STEP LENGTH
C
      DOPT = (XLK3 - 4.*XLK2 + 3.*XLK1)/(2.*(XLK3 - 2.*XLK2 + XLK1))
      IF(DOPT.GE.1.) DOPT = 1.
      IF(DOPT.LE.0.) DOPT = 0.5
      DELTA = DELTA*DOPT
C
C     SAVE NEW IMPROVED ESTIMATES
C
      DO 24 J = 1, 4
         B(J) = B(J) + DELTA*XTY(J)
         IF(J.NE.2) B(J)=AMAX1(B(J),0.)
         DO 27 K = 1, 4
            V(J,K) = XTX(J,K)
 27      CONTINUE
 24   CONTINUE
 20   CONTINUE
 23   CONTINUE
      XLK = XLK1
C
C     B  HAS FINAL ESTIMATES
C     XLK HAS LOG-LIKELIHOOD
C     V  HAS ESTIMATED COVARIANCE MATRIX OF PARAMETER ESTIMATES
C
      RETURN
      END
```

```
C     AMESTEST CURVE FIT - 3 PARAMETER MODEL          REVISED  6-24-80
C        PROGRAMMERS: HASSELBLAD, STEAD, CREASON
      SUBROUTINE AMT3
      DIMENSION XTY(4), XTX(4,4), D(4), BP(4), XL(50), XTXP(4,4),XTYP(4)
      COMMON N, X(200), Y(200), B(4), V(4,4), XLK, YFACT, NIT
C
C     N IS THE NUMBER OF COUNTS
C     X IS THE ARRAY OF DOSES
C     Y IS THE ARRAY OF COUNTS
C     B IS THE VECTOR OF PARAMETER ESTIMATES:
C        B(1) = BETA(0), B(2) = BETA(1), B(3) = BETA(2), B(4) = BETA(3)
C     V IS THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATES
C     XLK IS THE LIKELIHOOD OF THE CURRENT MODEL
C     YFACT IS THE SUM OF LOG(Y(I)!)
C     NIT IS THE NUMBER OF ITERATIONS FOR THE CURRENT MODEL
C
C     D IS THE VECTOR OF TERMS FOR THE 1ST PARTIALS AND THEIR SQUARES
C     XTX IS THE VECTOR OF 1ST PARTIALS
C     XTX IS THE MATRIX OF CROSS PRODUCTS OF 1ST PARTIALS
C     DELTA IS THE SCALAR WHICH DETERMINES THE LENGTH OF MOVEMENT
C     DOPT IS THE OPTIMAL LENGTH OF MOVEMENT
C
      NZ = 0
      B(1) = 0.
      BP(1) = 0.
      B(4) = 0.
      BP(4) = 0.
C
C     COMPUTE INITIAL ESTIMATES
C
      DO 2 I = 1, N
         IF(X(I).GT.0.) GO TO 2
         NZ = NZ + 1
         B(1) = B(1) + Y(I)
    2 CONTINUE
      B(1) = B(1)/FLOAT(NZ)
      DO 3 J = 1,2
         XTY(J) = 0.
         DO 4 K = 1,2
            XTX(J,K) = 0.
    4    CONTINUE
    3 CONTINUE
      DO 6 I = 1, N
      IF(X(I).LE.0.) GO TO 6
         D(1) = 1.
         D(2) = ALOG(X(I))
         YP = ALOG(AMAX1(5.,Y(I)-B(1)))
         DO 7 J = 1,2
            XTY(J) = XTY(J) + D(J)*YP
            DO 8 K = 1,2
```

```
                XTX(J,K) = XTX(J,K) + D(J)*D(K)
 8            CONTINUE
 7          CONTINUE
 6        CONTINUE
          CALL INVSPD(XTX,XTY,2,DET)
C
C         CONSTRAIN INITIAL ESTIMATES TO BE POSITIVE
C
          B(2) = XTY(1)
          B(3) = AMAX1(0.,XTY(2))
C
C         ITERATE USING A GAUSS-NEWTON METHOD TO FIND THE MAXIMUM
C         LIKELIHOOD SOLUTION
C
          DO 20 IT = 1, 500
          NIT = IT
          XLK1 = -YFACT
          DELTA = 1.
C
C         COMPUTE PREDICTED LAMBDAS, CURRENT LIKELIHOOD
C
          DO 9 I = 1, N
             XL(I) = B(1)
             IF(X(I).GT.0.) XL(I) = (B(1)+EXP(B(2)+B(3)*ALOG(X(I))))
             IF(XL(I).EQ. 0.) XL(I) = 1.E-18
             XLK1 = XLK1 + Y(I)*ALOG(XL(I)) - XL(I)
 9        CONTINUE
          DO 11 J = 1, 4
             XTY(J) = 0.
             DO 12 K = 1, 4
                XTX(J,K) = 0.
 12          CONTINUE
 11       CONTINUE
C
C         COMPUTE ELEMENTS OF FIRST PARTIAL DERIVATIVES
C         SUM THESE AND THEIR CROSS PRODUCTS
C
          DO 14 I = 1, N
             TERM = Y(I)/XL(I) - 1.
             D(1) = TERM
             D(2) = 0.
             IF(X(I).GT.0.) D(2)=TERM*EXP(B(2)+B(3)*ALOG(X(I)))
             D(3) = 0.
             IF(X(I).GT.0.) D(3) = ALOG(X(I))*EXP(B(2)+B(3)*ALOG(X(I)))
          1   *TERM
             DO 15 J = 1,3
                XTY(J) = XTY(J) + D(J)
                DO 16 K = 1,3
                   XTX(J,K) = XTX(J,K) + D(J)*D(K)
 16             CONTINUE
```

```
 15      CONTINUE
 14   CONTINUE
C
C     COMPUTE CHANGE IN ESTIMATES
C
      CALL INVSPD(XTX,XTY,3,DET)
C
C     COMPUTE CONSTRAINED NEW ESTIMATES, AND TRY THEM
C
      DO 17 J = 1,3
         BP(J) = B(J) + XTY(J)
         IF(J.NE.2) BP(J)=AMAX1(BP(J),0.)
 17   CONTINUE
      XLK2 = -YFACT
      DO 18 I = 1, N
         XL(I) = BP(1)
         IF(X(I).LE.0.) GO TO 28
         TEMP = BP(2) + BP(3)*ALOG(X(I))
         IF(TEMP.GT.50.) TEMP = 50.
         XL(I) = BP(1) + EXP(TEMP)
 28      CONTINUE
         IF(XL(I).EQ.0.) XL(I) = 1.E-18
         XLK2 = XLK2 + Y(I)*ALOG(XL(I)) - XL(I)
 18   CONTINUE
C
C     FIND BEST ESTIMATES USING A CHOPPING METHOD
C
 19   CONTINUE
      XLK3 = XLK2
      DELTA = DELTA*0.5
C
C     IF CHOPPING GETS TOO SMALL, QUIT
C
      IF(DELTA.LE.0.00001) GO TO 23
      DO 21 J = 1,3
         BP(J) = B(J) + DELTA*XTY(J)
         IF(J.NE.2) BP(J)=AMAX1(BP(J),0.)
 21   CONTINUE
      XLK2 = -YFACT
      DO 22 I = 1, N
         XL(I) = BP(1)
         IF(X(I).LE.0.) GO TO 29
         TEMP = BP(2) + BP(3)*ALOG(X(I))
         IF(TEMP.GT.50.) TEMP = 50.
         XL(I) = BP(1) + EXP(TEMP)
 29      CONTINUE
         IF(XL(I).EQ.0.) XL(I) = 1.E-18
         XLK2 = XLK2 + Y(I)*ALOG(XL(I)) - XL(I)
 22   CONTINUE
      IF(XLK2.LE.XLK1) GO TO 19
```

```
C
C     OPTIMIZE STEP LENGTH
C
      DOPT = (XLK3 - 4.*XLK2 + 3.*XLK1)/(2.*(XLK3 - 2.*XLK2 + XLK1))
      IF(DOPT.GE.1.) DOPT = 1.
      IF(DOPT.LE.0.) DOPT = 0.5
      DELTA = DELTA*DOPT
C
C     SAVE NEW IMPROVED ESTIMATES
C
      DO 24 J = 1,3
         B(J) = B(J) + DELTA*XTY(J)
         IF(J.NE.2) B(J)=AMAX1(B(J),0.)
         DO 27 K = 1, 3
            V(J,K) = XTX(J,K)
 27      CONTINUE
 24   CONTINUE
 20   CONTINUE
 23   CONTINUE
      XLK = XLK1
C
C     B  HAS FINAL ESTIMATES
C     XLK HAS LOG-LIKELIHOOD
C     V  HAS ESTIMATED COVARIANCE MATRIX OF PARAMETER ESTIMATES
C
      RETURN
      END
```

```
C     AMESTEST CURVE FIT - 2 PARAMETER MODEL          REVISED  5-19-80
C        PROGRAMMERS: HASSELBLAD, STEAD, CREASON
C
      SUBROUTINE AMT2
      COMMON N, X(200), Y(200), B(4), V(4,4), XLK, YFACT, NIT
C
C     N IS THE NUMBER OF COUNTS
C     X IS THE ARRAY OF DOSES
C     Y IS THE ARRAY OF COUNTS
C     B IS THE VECTOR OF PARAMETER ESTIMATES:
C        B(1) = BETA(0), B(2) = BETA(1), B(3) = BETA(2), B(4) = BETA(3)
C     V IS THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATES
C     XLK IS THE LIKELIHOOD OF THE CURRENT MODEL
C     YFACT IS THE SUM OF LOG(Y(I)!)
C     NIT IS THE NUMBER OF ITERATIONS FOR THE CURRENT MODEL
C
C     D1, D2 IS THE VECTOR OF TERMS FOR THE 1ST PARTIALS
C     D11,D12,D22 IS THE MATRIX OF CROSS PRODUCTS OF 1ST PARTIALS
C     DET IS THE DETERMINANT OF THE MATRIX
C     CH1,CH2 ARE THE CHANGES IN B(1) AND B(4) RESPECTIVELY
C
      CN = N
      XS = 0.
      X2S = 0.
      XYS = 0.
      YS = 0.
C
C     COMPUTE INITIAL ESTIMATES
C
      DO 2 I = 1, N
      YL = ALOG(AMAX1(Y(I),5.))
      XS = XS + X(I)
      X2S = X2S + X(I)**2
      XYS = XYS + X(I)*YL
      YS = YS + YL
 2    CONTINUE
      DET = CN*X2S - XS**2
      B(1) = (X2S*YS - XS*XYS)/DET
      B(2) = 0.
      B(3) = 0.
      B(4) = (CN*XYS - XS*YS)/DET
C
C     ITERATE USING A NEWTON RAPHSON METHOD
C
      DO 4 IT = 1, 200
      NIT = IT
      D1 = 0.
      D2 = 0.
      D11 = 0.
      D12 = 0.
```

```
      D22 = 0.
      DO 3 I = 1, N
         XLM = EXP(B(1) + B(4)*X(I))
         T1 = Y(I) - XLM
         D1 = D1 + T1
         D2 = D2 + X(I)*T1
         D11 = D11 + XLM
         D12 = D12 + XLM*X(I)
         D22 = D22 + XLM*X(I)**2
 3    CONTINUE
      DET = D11*D22 - D12**2
      CH1 = (D1*D22 - D2*D12)/DET
      CH2 = (D2*D11 - D1*D12)/DET
      B(1) = B(1) + CH1
      B(4) = B(4) + CH2
C
C     TEST FOR CONVERGENCE
C
      IF(ABS(CH1/B(1),  ABS(CH2/B(4)).LT.0.0001) GO TO 5
 4    CONTINUE
 5    CONTINUE
      XLK = -YFACT
      DO 6 I = 1, N
         XLM = EXP(B(1) + B(4)*X(I))
         XLK = XLK - XLM + Y(I)*ALOG(XLM)
 6    CONTINUE
      B(4) = - B(4)
      IF(B(4).LT.0.) B(1) = YS/CN
      IF(B(4).LT.0.) B(4) = 0.
      RETURN
      END
```

```
C     PLOTIT                                                REVISED  5-21-80
C          PROGRAMMERS: CREASON, STEAD
      SUBROUTINE PLOTIT
C
C     PLOTS THE OBSERVED COUNTS VS. THE DOSE, AND DRAWS THE ESTIMATED
C     CURVE
C
      DIMENSION NTITL(18),X2(100),Y2(100)
C
C     NTITL IS THE ARRAY USED TO SET UP THE TITLE FOR THE GRAPH
C     X2 IS THE ARRAY OF COMPUTED DOSE VALUES FOR THE FITTED CURVE
C     Y2 IS THE ARRAY OF COMPUTED FUNCTION VALUES FOR THE FITTED CURVE
C
      COMMON N,X(200),Y(200),B(4),V(4,4),XLK,YFACT,NIT
C
C     N IS THE NUMBER OF COUNTS
C     X IS THE ARRAY OF DOSES
C     Y IS THE ARRAY OF COUNTS
C     B IS THE VECTOR OF PARAMETER ESTIMATES:
C        B(1) = BETA(0), B(2) = BETA(1), B(3) = BETA(2), B(4) = BETA(3)
C     V IS THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATES
C     XLK IS THE LIKELIHOOD OF THE CURRENT MODEL
C     YFACT IS THE SUM OF LOG(Y(I)!)
C     NIT IS THE NUMBER OF ITERATIONS FOR THE CURRENT MODEL
C
      DATA NTITL /79,66,83,32,38,32,69,88,80,32,86,83,
     $ 32,68,79,83,69,32/
C
C     COMPUTE EXPECTED VALUES FROM THE MODEL
C
      XMIN = X(1)
      XMAX = X(1)
      YMIN = Y(1)
      YMAX = Y(1)
      DO 4 I = 2, N
         IF(X(I) .LT. XMIN) XMIN = X(I)
         IF(X(I) .GT. XMAX) XMAX = X(I)
 4    CONTINUE
      XINC = (XMAX - XMIN)/100.0
      DO 3 I = 1, 100
         X2(I) = XMIN + (I-1)*XINC
         Y2(I) = B(1)*EXP(-B(4)*X2(I))
         IF(X2(I) .LE. 0.0) GO TO 2
         Y2(I) = Y2(I) + EXP(B(2)+B(3)*ALOG(X2(I))-B(4)*X2(I))
 2    CONTINUE
      IF(Y2(I) .LT. YMIN) YMIN = Y2(I)
      IF(Y2(I) .GT. YMAX) YMAX = Y2(I)
 3    CONTINUE
C
C     SET UP PLOT-10 PARAMETERS
```

```
C
C
C     DRAW PLOT WITH FITTED MODEL
C
      CALL MOVABS(730,400)
      CALL HLABEL(18,NTITL)
      CALL SLIMY(40,390)
      CALL SLIMX(675,1000)
      CALL DLIMX(XMIN,XMAX)
      CALL DLIMY(YMIN,YMAX)
      CALL FRAME
      CALL SYMBL(2)
      CALL LINE(-1)
      CALL NPTS(N)
      CALL CHECK(X,Y)
      CALL DSPLAY(X,Y)
      CALL LINE(0)
      CALL SYMBL(0)
      CALL NPTS(100)
      CALL CPLOT(X2,Y2)
      CALL HDCOPY
      CALL ERASE
      CALL FINITT(0,380)
      RETURN
      END
```

```
C      PRINTER PLOT ROUTINE                                    REVISED  6-27-80
C         PROGRAMMERS: DUNCAN, HASSELBLAD
       SUBROUTINE PRPLT(NUNP)
       DIMENSION XSCL(6), YP(101)
       COMMON N, X(200), Y(200), B(4), V(4,4), XLK, YFACT, NIT
C
C      N IS THE NUMBER OF COUNTS
C      X IS THE ARRAY OF DOSES
C      Y IS THE ARRAY OF COUNTS
C      B IS THE VECTOR OF PARAMETER ESTIMATES:
C         B(1) = BETA(0), B(2) = BETA(1), B(3) = BETA(2), B(4) = BETA(3)
C      NTR IS AN ARRAY OF VALUES TO LABEL THE HORIZONTAL AXIS
C      NPL IS THE ARRAY OF CHARACTERS FORMING THE PLOT
C      NUNP IS THE 3 CHARACTER UNITS LABEL FROM THE MAIN PROGRAM
C
       CHARACTER*1 NPL(51,101),NBL,NP,NMN,NPD,NSK,LX,NCL,LBL(51)
       DATA NBL,NP,NMN,NPD,NSK,LX,NCL/' ','+','-','.','*','X',':'/
       DATA LBL/8*' ','O','B','S','E','R','V','E','D',' ','C','O','U','N'
      1,'T','S',' ','A','N','D',' ','P','R','E','D','I','C','T','E','D',
      2 ' ','C','U','R','V','E',8*' '/
C
C      COMPUTE MAXIMUM X, Y
C
       XMAX = X(1)
       YMAX = Y(1)
       DO 2 I = 2, N
          IF(X(I).GT.XMAX) XMAX = X(I)
          IF(Y(I).GT.YMAX) YMAX = Y(I)
  2    CONTINUE
       XMAXD = XMAX
C
C      COMPUTE MAXIMUM X FOR PLOT
C
       XSC = ALOG10(XMAX) + 10.
       NXSC = XSC - 0.01
       XMAX = 2.5*10.**(NXSC - 10)
       XSC = XSC - NXSC
       IF(XSC.GT.0.39794) XMAX = XMAX*2.
       IF(XSC.GT.0.69897) XMAX = XMAX*2.
C
C      COMPUTE FITTED CURVE
C
       YP(1) = B(1)
       NMAX = 1
       DO 8 J = 2, 101
          XX = (J - 1)*XMAX/100.
          IF(XX.GT.XMAXD) GO TO 14
          NMAX = J
          YP(J) = (B(1)+EXP(B(2)+B(3)*ALOG(XX)))*EXP(-B(4)*XX)
          IF(YP(J).GT.YMAX) YMAX = YP(J)
```

```
 8    CONTINUE
 14   CONTINUE
C
C     COMPUTE MAXIMUM Y FOR PLOT
C
      XSC = ALOG10(YMAX) + 10.
      NXSC = XSC - 0.01
      XSC = XSC - NXSC
      YMAX = 2.5*10.**(NXSC - 10)
      IF(XSC.GT.0.39794) YMAX = YMAX*2.
      IF(XSC.GT.0.69897) YMAX = YMAX*2.
      IYY = YMAX/5. + .01
C
C     INITIALIZE THE PLOT ARRAY
C
      DO 3 I = 1, 51
         II = ((I - 1)/10)*10 + 1
         DO 4 J = 2, 100
            NPL(I,J) = NBL
 4       CONTINUE
         IF(I.EQ.II) NPL(I,1) = NP
         IF(I.EQ.II) NPL(I,101) = NP
         IF(I.NE.II) NPL(I,1) = NCL
         IF(I.NE.II) NPL(I,101) = NCL
 3    CONTINUE
      DO 6 J = 1, 101
         JJ = ((J - 1)/10)*10 + 1
         IF(J.EQ.JJ) NPL(1,J) = NP
         IF(J.EQ.JJ) NPL(51,J) = NP
         IF(J.NE.JJ) NPL(1,J) = NMN
         IF(J.NE.JJ) NPL(51,J) = NMN
 6    CONTINUE
C
C     STORE THE FITTED CURVE
C
      DO 16 J = 1, NMAX
         IY = (YP(J)/YMAX)*50. + 1.5
         IY = 52 - IY
         NPL(IY,J) = NPD
 16   CONTINUE
C
C     STORE THE OBSERVED POINTS
C
      DO 7 I = 1, N
         IX = (X(I)/XMAX)*100. + 1.5
         IY = (Y(I)/YMAX)*50. + 1.5
         IY = 52 - IY
         IF(NPL(IY,IX).EQ.LX) GO TO 12
         NPL(IY,IX) = LX
         GO TO 7
```

```
 12   CONTINUE
         NPL(IY,IX) = NSK
 7    CONTINUE
      WRITE (6,5)
 5    FORMAT(' ')
C
C     WRITE OUT THE PLOT
C
      DO 9 I = 1, 51
         IL = (I - 1)/10
         IW = (5 - IL)*IYY
         II = IL*10 + 1
         IF(I.EQ.II) WRITE (6,10) LBL(I), IW, (NPL(I,J),J=1,101)
 10      FORMAT(7X,A1,I6,101A1)
         IF(I.NE.II) WRITE (6,15) LBL(I), (NPL(I,J),J=1,101)
 15      FORMAT(7X,A1,6X,101A1)
 9    CONTINUE
      DO 13 I = 1, 6
      XSCL(I) = (XMAX/5.)*(I-1)
 13   CONTINUE
      WRITE (6,20) XSCL
 20   FORMAT(5X,F10.2,5(10X,F10.2))
      WRITE (6,25) NUNP
 25   FORMAT('0',55X,'DOSE (',A3,')')
      RETURN
      END
```

```
C     INVSPD                                                REVISED  6-18-80
C        PROGRAMMER: BRANTON
      SUBROUTINE INVSPD(A,B,N,DET)
C
C     ***********************************************************
C     *                                                         *
C     *   THIS SUBROUTINE COMPUTES THE INVERSE OF A SYMMETRIC*
C     * POSITIVE-DEFINITE MATRIX. THE DETERMINANT OF THE      *
C     * MATRIX AND THE SOLUTION VECTOR FOR THE SYSTEM OF      *
C     * SIMULTANEOUS EQUATIONS CORRESPONDING TO THIS MATRIX   *
C     * ARE COMPUTED.                                         *
C     *   THE SUBROUTINE IS PASSED A MATRIX, A, A VECTOR, B, *
C     * (CORRESPONDING TO THE EQUATION AX=B), AND AN INTEGER,*
C     * N, THE ROW (OR COLUMN) DIMENSION OF A. THE INVERSE OF*
C     * A IS RETURNED IN PLACE OF A, THE SOLUTION VECTOR IN   *
C     * PLACE OF B, AND THE DETERMINANT. IF  A IS NOT INVERT-*
C     * IBLE, CONTROL IS RETURNED TO THE CALLING PROGRAM.     *
C     *   N SHOULD BE A POSITIVE INTEGER, A SHOULD BE AN NXN *
C     * REAL MATRIX, B, COLMS & SCALE  N-DIMENSIONAL          *
C     * VECTORS, AND DET  REAL.                               *
C     *                                                       *
C     ***********************************************************
C
      DIMENSION A(4,4),B(4),COLMS(4),SCALE(4)
      INTEGER COL,COLMS
C
C     INITIALIZATION
C
      DET=1.
      DO 1 I=1,N
         COLMS(I)=I
         SCALE(I)=0.
         DO 4 J=1,N
            SCALE(I)=SCALE(I)+ABS(A(I,J))
    4    CONTINUE
    1 CONTINUE
C
C     SET PIVOT FOR ITH COL. & PERFORM REDUCTION
C
      DO 30 ICOL=1,N
         PIVOT=A(COLMS(ICOL),COLMS(ICOL))
         DO 2 J=ICOL,N
            IF(ABS(PIVOT/SCALE(COLMS(ICOL))).GE.
     *           ABS(A(COLMS(J),COLMS(J))/SCALE(COLMS(J))))GO TO 2
            PIVOT=A(COLMS(J),COLMS(J))
            SWAP=COLMS(J)
            COLMS(J)=COLMS(ICOL)
            COLMS(ICOL)=SWAP
    2    CONTINUE
    3    COL=COLMS(ICOL)
C
C        IF PIVOT=0 THEN THE MATRIX IS SINGULAR
C        CONTROL IS RETURNED TO THE CALLING PROGRAM
```

```
C
   35    IF(PIVOT.NE.0.)GO TO 55
         RETURN
C
C        UPDATE THE DETERMINATE
C
   55    DET=DET*PIVOT
C
C        PERFORM THE REDUCTION
C
C        FIRST THE PIVOT ROW
C
         A(COL,COL)=1.
         DO 60 J=1,N
            A(COL,J)=A(COL,J)/PIVOT
   60    CONTINUE
C
C        NOW B
C
         B(COL)=B(COL)/PIVOT
C
C        NEXT, THE NON-PIVOT ROWS
C
         DO 70 I=1,N
            IF(I.EQ.COL)GO TO 70
            TEMP=A(I,COL)
            A(I,COL)=0.
            DO 80 J=1,N
               A(I,J)=A(I,J)-A(COL,J)*TEMP
   80       CONTINUE
C
C           ADJUST B
C
            B(I)=B(I)-B(COL)*TEMP
   70    CONTINUE
C
C        RESCALE
C
         IF(ICOL.GE.N-1)GO TO 30
         DO 75 J=ICOL+1,N
            SCALE(COLMS(J))=0.
            DO 75 K=ICOL+1,N
               SCALE(COLMS(J))=SCALE(COLMS(J))+ABS(A(COLMS(J),COLMS(K)))
   76       CONTINUE
   75    CONTINUE
C
C     THE ITH COL IS DONE WITH. RETURN TO THE TOP OF THE
C     LOOP TO PROCESS THE NEXT COLUMN
C
   30 CONTINUE
C
      RETURN
      END
```

```
C     INTRP                                                  REVISED  5-19-80
C          PROGRAMMER: HASSELBLAD
      SUBROUTINE INTRP(NCD,NTRP,K)
C
C     CONVERTS A STRING OF ALPHABETIC NUMBERS TO AN ARRAY OF INTEGERS
C     NCD IS THE INPUT ARRAY OF ALPHABETIC NUMBERS
C     NTRP IS THE ARRAY OF INTEGERS FOUND
C     K IS THE NUMBER OF INTEGERS FOUND
C
      DIMENSION NCHR(10), NCD(25), NTRP(10)
      DATA NCHR/'1','2','3','4','5','6','7','8','9','0'/
      DATA NB/' '/
      N = 0
      K = 0
      NF = 0
      DO 3 I = 1, 10
         NTRP(I) = 0
 3    CONTINUE
      DO 2 I = 1, 25
         DO 4 J = 1, 10
C
C           CHECK FOR RECOGNIZABLE CHARACTER
C
            IF(NCD(I).NE.NCHR(J)) GO TO 4
            IF(I.EQ.1) GO TO 6
C
C           CHECK FOR PRECEEDING BLANK
C
            IF(NF.EQ.0.AND.NCD(I-1).NE.NB) GO TO 4
 6          CONTINUE
            IF(NF.EQ.0) NF = 1
            L = J
            IF(J.EQ.10) L = 0
            N = N*10 + L
            GO TO 2
 4       CONTINUE
         IF(NF.EQ.0) GO TO 2
         K = K + 1
         NTRP(K) = N
         N = 0
         NF = 0
         IF(K.GE.10) RETURN
 2    CONTINUE
      IF(NF.EQ.0) RETURN
      K = K + 1
      NTRP(K) = N
      RETURN
      END
```

```
C     CHISQ                                                REVISED  5- 9-80  CHISQ001
C        PROGRAMMER: HASSELBLAD                                             CHISQ002
C     COMPUTES THE CHI-SQUARE INTEGRAL TO X FOR N DEGREES OF FREEDOM        CHISQ003
C        REFERENCE: HANDBOOK OF MATHEMATICAL FUNCTIONS                      CHISQ004
C                                                                           CHISQ005
      REAL FUNCTION CHISQ(X,N)                                              CHISQ006
      SUM = 0.                                                              CHISQ007
      IF(X.GT.0.0.AND.N.GT.0)GO TO 2                                        CHISQ008
      CHISQ = 1.                                                            CHISQ009
      RETURN                                                                CHISQ010
 2    CONTINUE                                                              CHISQ011
      IF(N.GT.50) GO TO 6                                                   CHISQ012
      IF( (N/2)*2.NE.N)GO TO 11                                             CHISQ013
      IF(N.LE.2)GO TO 3                                                     CHISQ014
      M = N/2 - 1                                                           CHISQ015
      TERM = 1.                                                             CHISQ016
      DO 4 I = 1, M                                                         CHISQ017
      CI = 2.*FLOAT(I)                                                      CHISQ018
      TERM = TERM*X/CI                                                      CHISQ019
      SUM = SUM + TERM                                                      CHISQ020
 4    CONTINUE                                                              CHISQ021
 3    CONTINUE                                                              CHISQ022
      CHISQ = EXP(-X/2.0)*(1.0+SUM)                                         CHISQ023
      RETURN                                                                CHISQ024
 11   CONTINUE                                                              CHISQ025
      TERM = 1.0/X                                                          CHISQ026
      IF(N.LE.1)GO TO 12                                                    CHISQ027
      M = (N-1)/2                                                           CHISQ028
      DO 13 I = 1, M                                                        CHISQ029
      CI = 2.*FLOAT(I)-1.                                                   CHISQ030
      TERM = TERM*X/CI                                                      CHISQ031
      SUM = SUM + TERM                                                      CHISQ032
 13   CONTINUE                                                              CHISQ033
 12   CONTINUE                                                              CHISQ034
      XS = SQRT(X)                                                          CHISQ035
      CHISQ = 2. - 2.*CNORM(XS) + XS*EXP(-X/2.)*.7978846*SUM                CHISQ036
      RETURN                                                                CHISQ037
 6    CONTINUE                                                              CHISQ038
      CHISQ = 1.-CNORM(((X/N)**.3333333-1.+2./(9.*N))/SQRT(2./(9.*N)))      CHISQ039
      RETURN                                                                CHISQ040
      END                                                                   CHISQ041
```

```
C      CNORM                                                  REVISED  5- 9-80 CNORM001
C         PROGRAMMER: HASSELBLAD                                               CNORM002
C      COMPUTES THE NORMAL PROBABILITY INTEGRAL FROM MINUS INFINITY TO X       CNORM003
C      REFERENCE: APPROX. FOR DIG. COMP. - HASTINGS, P. 169                    CNORM004
       FUNCTION CNORM(X)                                                       CNORM005
       DATA S2P,P,A1,A2,A3,A4,A5,S2/1.12837917 ,.3275911,.225836846,           CNORM006
      1  -.252128668,1.25969513,-1.28782245 ,.94064607,.707106781/             CNORM007
       XN = 1./(1. + P*ABS(X*S2))                                              CNORM008
       PHI = S2P*EXP(-X**2/2.)                                                 CNORM009
       PHIS=1.-XN*(A1+XN*(A2+XN*(A3+XN*(A4+XN*A5))))*PHI                       CNORM010
       TERM = .5*PHIS                                                          CNORM011
       IF(X.LT.0.) TERM = -TERM                                                CNORM012
       CNORM = 0.5 + TERM                                                      CNORM013
       RETURN                                                                  CNORM014
       END                                                                     CNORM015
```

```
C      ALGMA                                                  REVISED  5- 9-80 ALGMA001
C         PROGRAMMER: HASSELBLAD                                               ALGMA002
C      COMPUTES LOG(GAMMA(X)) USING STIRLING'S APPROXIMATION                   ALGMA003
C                                                                              ALGMA004
       REAL FUNCTION ALGMA(X)                                                  ALGMA005
       Z = X                                                                   ALGMA006
       SUM = 0.                                                                ALGMA007
       IF(X.LE.0.) GO TO 9                                                     ALGMA008
 3     CONTINUE                                                                ALGMA009
       IF(Z.GE.5.) GO TO 2                                                     ALGMA010
       SUM = SUM - ALOG(Z)                                                     ALGMA011
       Z = Z + 1                                                               ALGMA012
       GO TO 3                                                                 ALGMA013
 2     CONTINUE                                                                ALGMA014
       TERM = 1. + .083333333/Z + .0034722222/Z**2 - .0026813272/Z**3          ALGMA015
       ALGMA = (Z - .5)*ALOG(Z) - Z + .91893853 + ALOG(TERM) + SUM             ALGMA016
       RETURN                                                                  ALGMA017
 9     CONTINUE                                                                ALGMA018
       ALGMA = -0.                                                             ALGMA019
       RETURN                                                                  ALGMA020
       END                                                                     ALGMA021
```

11. Results of Simulation Runs

Simulations were run to evaluate the performance of this program especially as relating to bias in the estimation procedure. Random Poisson counts were generated to simulate data which might have arisen under typical experimental conditions of Ames testing. The six dose levels chosen were 0.00, 0.25, 0.50, 0.75, 2.50, and 7.50 units. Two counts were generated at each dose level. One thousand of these 12-count data sets were created per simulation.

Original Parameterization of the Four Parameter Model

Initially, the counts were generated with the Poisson* parameter λ modeled as:

$$\lambda_i(x_i) = \beta_0, \text{ if } x_i=0$$

$$= [\beta_0 + \beta_1 \exp(\beta_2 \ln(x_i))]\exp(-\beta_2 x_i) \text{ if } x_i>0 \qquad (11.1)$$

were β_0, β_1, β_2 and β_3 are constrained to be positive, and x_i represents the ith dose level. Note that the definition of β_1 is different from that in section 1. The Poisson distribution of probabilities have the form

$$P(x) = e^{-\lambda}\lambda^x/x! \qquad (11.2)$$

Individual simulations were conducted for each of the four following sets of values of β_0, β_1, β_2 and β_3:

Simulation	β_0	β_1	β_2	β_3
A	30.1966	.3421	1.2517	0.0000
B	30.1966	.3421	1.2517	0.0005
C	30.1966	.3421	1.2517	0.0010
D	30.1966	.3421	1.2517	0.0030

So, for example, in simulation A and simulation C, the values of $\lambda_i(x_i)$ were:

Simulation	0.00	0.25	0.50	0.75	2.5	7.5
A	30.19660	30.25693	30.34027	30.43525	31.27370	34.45717
C	30.1966 0	30.25692	30.34019	30.43507	31.27101	34.42533

For each set of 12 counts, an iterative procedure similar described for the AMT4 subroutine (section 5) was used to obtain the (biased) maximum likelihood estimates of β_0, β_1, β_2 and β_3 for the model (11.1). The averages of the 1000 values of $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\beta}_3$ were obtained for each simulation:

Averages of the parameter estimates

Simulation	β_0	β_1	β_2	β_3
A	32.861	0.215	1.437	0.0007
B	31.301	0.309	1.341	0.0008
C	30.888	0.356	1.296	0.0012
D	30.605	0.387	1.270	0.0031

These estimates are relatively close to the parameter values. There appears to be more bias for the smaller values of β_3, which supports the use of the 3 parameter model where β_3 is near zero.

Reparameterized Four Parameter Model

The model was reparameterized into the form shown in section 1

$$\lambda_i(x_i) = \beta_0, \qquad x_i=0$$

$$= (\beta_0 + \exp(\beta_1 + \beta_2 \ln(x_i))\exp(-\beta_3 x_i) \qquad x_i>0 \qquad (11.3)$$

where β_1 is not constrained to be positive. The same counts used in simulation C were used for this model. Simulation C was chosen because it had a relatively small value of β_3. Average values of the 1000 estimates of β_0, β_1, β_2 and β_3 were 30.720, -1.134, 1.265, and .00104 respectively. Note that for this parameterization, the true value of β_1 is -1.073.

The results indicate that the reparameterized model gives average estimates which are closer to the true parameter values.

In addition, the number of iterations required for convergence of the estimates for β_0, β_1, β_2 and β_3, decreased substantially. Whereas under the original parameterization, only 10% of the 1000 estimates were obtained in 60 or fewer iterations, with the final parameterization 60% converged within 60 iterations. And whereas in the original parameterization 47% did not converge to an estimate within the limits of the program, ie., within 200 iterations, in the new parameterization, only 4% did not converge. Finally, the sample variances of the 1000 estimates were slightly smaller for the reparameterized model. Thus the reparameterized model provided better estimates for the same observed counts.

Estimates of the Variances and Correlations

The matrix W^{-1} (as defined in the AMT4 subroutine, section 5) is an estimate of the covariance of the parameter estimates. Let $V = W^{-1}$ so that V_{ii} is an estimate of the asymptotic variance of β_i, i=0, 1, 2, 3. The average of these estimated variances can be compared with the observed sample variances

	parameter			
	β_0	β_1	β_2	β_3
Average estimated variance	104.67	1.559	.074	$.813X10^{-6}$
Sample variance	15.81	.363	.017	$.120X10^{-6}$

Thus, the estimated variances tend to be significantly larger than the observed variances. For this reason, it was decided not to use these estimated variances for tests of significance.

Several pairs of the parameter estimates were highly correlated. For simulation C, the following is the estimated correlation matrix:

	β_0	β_1	β_2	β_3
β_0	1.00	-.592	.555	.450
β_1	-.592	1.000	-.996	-.942
β_2	.555	-.996	1.000	.967
β_3	.450	-.942	.967	1.000

Figures 11.1, 11.2, and 11.3 show some of these relationships. The first two figures show a clearly nonlinear component, indicating that the associations are even stronger than indicated by the magnitudes of the correlations. This also gives some indication of the difficulty in the estimation procedure when the value of β_3 is small.

Tests for Toxicity

Simulation A was run with no toxicity (β_3=0). This same data was fitted using the three parameter model. The statistic

$$X^2 = -2(\ln(L_2)-\ln(L_1))$$

where L_2 and L_1 are the likelihoods of the 3 and 4 parameter models, respectively, is asymptotically distributed as a chi-square with one degree of freedom. This test for toxicity under the null hypothesis was compared with the percentiles of the actual chi-square distribution.

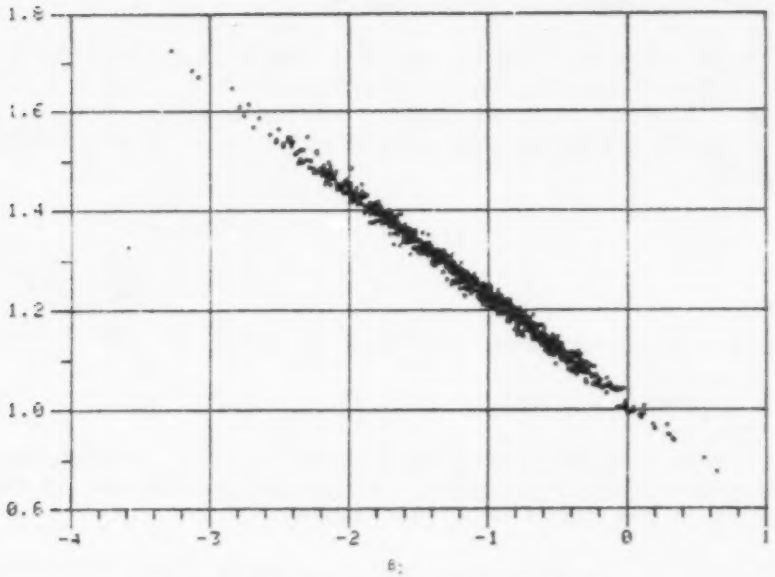


Figure 11.1 Plot of β_2 vs. β_1 for 1000 simulations

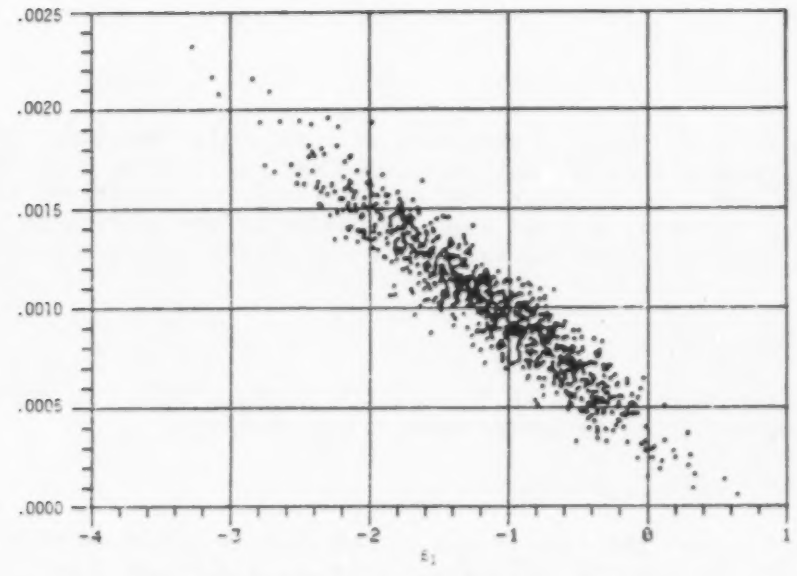


Figure 11.2 Plot of β_3 vs. β_1 for 1000 simulations

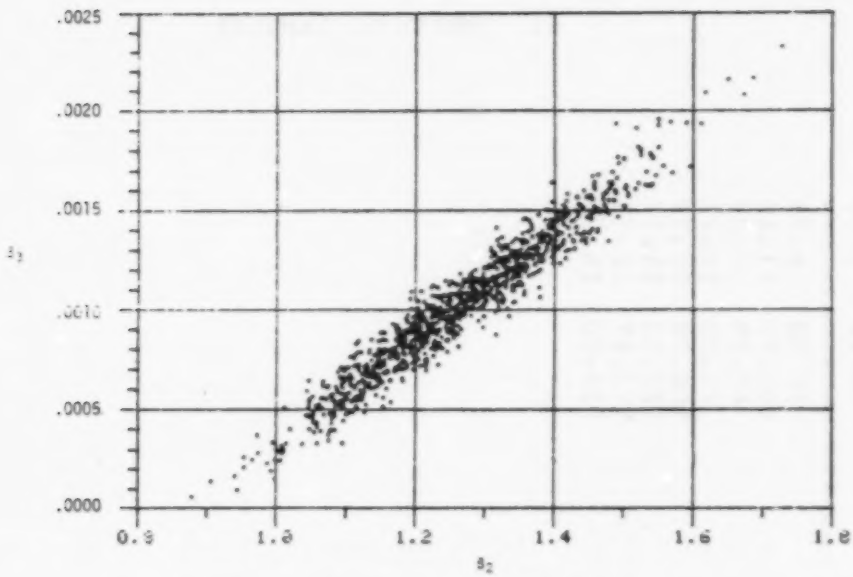


Figure 11.3 Plot of β_3 vs. β_2 for 1000 simulations

	percentiles			
	.75	.90	.95	.99
Observed X statistic:	.4945	1.584	2.654	4.597
Actual chi-square:	1.323	2.706	3.841	6.635

Thus the X^2 statistic appears to be quite conservative, that is, it is less likely to make a type I error than would be expected. One reason for this might be the computational difficulties for the 4 parameter model when $\beta_3=0$. Several of the observed X^2 were actually negative, when they should have been positive. These values probably would have been small values, and thus the decision to use the 3 parameter model would be valid even though the 4 parameter model failed to converge properly.

SAMPLE ID: GTDX790002NSNC ACTIVATION: +
STRAIN: TA98 DATE: 4/19/79 TECHNICIAN: PMM

DOSE	UNITS	PLATE COUNTS			MEAN	S.D.
.00	UGS	39	33		36.00	4.24
.50*	UGS	483	528		505.50	31.82
25.00	UGS	54	40	47	47.00	7.00
75.00	UGS	101	45	67	71.00	28.21
125.00	UGS	188	170	149	169.00	19.52
250.00	UGS	291	315	361	322.33	35.57
750.00	UGS	597	624	560	593.67	32.13
1250.00	UGS	746	750	703	733.00	26.06

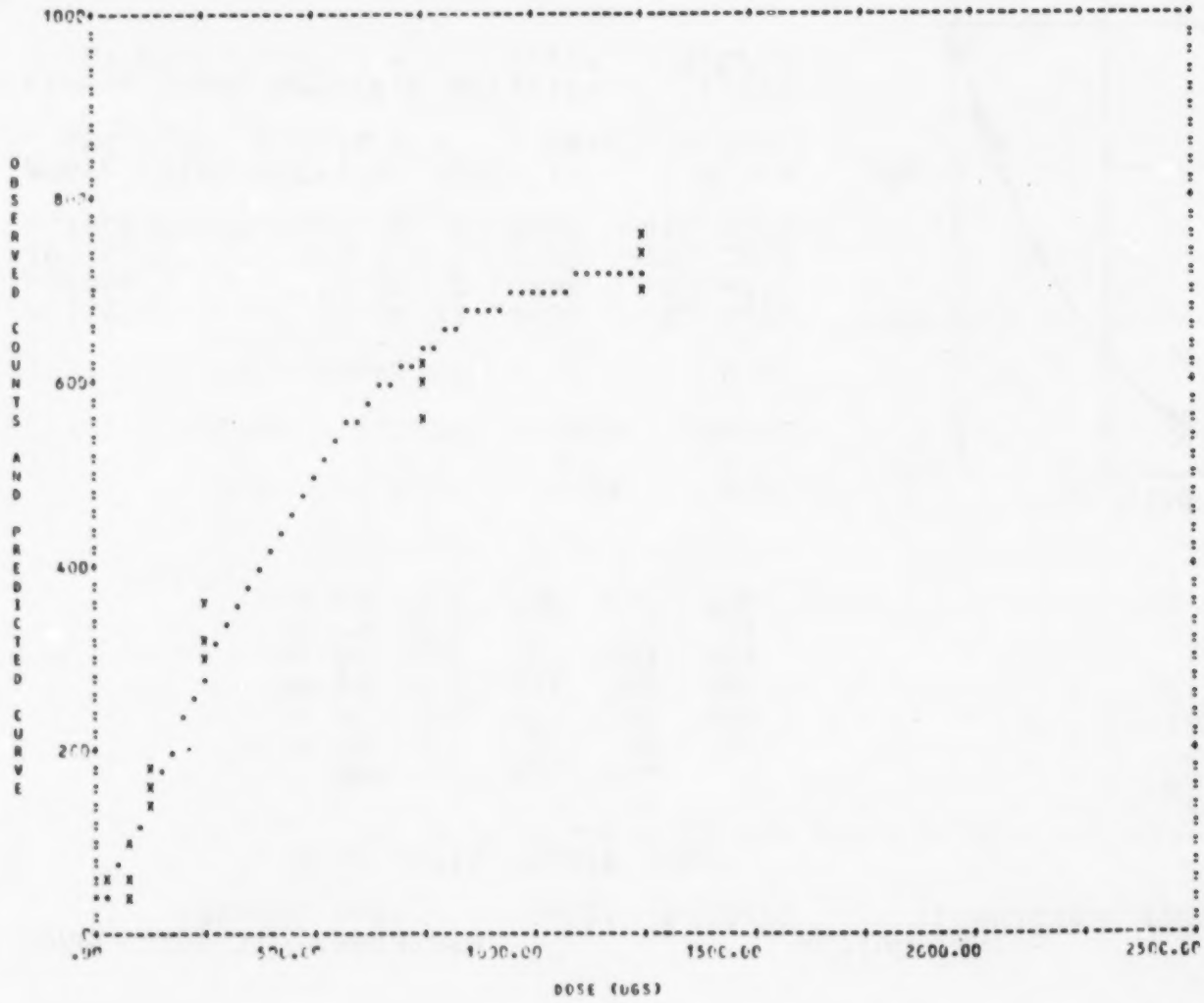
	B(0)	B(1)	B(2)	B(3)
ESTS.	30.195	-1.0710	1.2515	.00104

TEST	CHI-SQUARE	DF	P	LOGL
POISSON	42.70	13	.0001	-91.1926
ADEQUACY	54.97	3	.0000	-118.6781
TOXICITY	112.13	1	.0000	-174.7446
MUTAGENICITY	1029.20	2	.0000	-633.2803

AVERAGE SLOPE (NONLIN. MODEL) = 2.059
95% CONF. LIMITS = (1.787, 2.373)

AVERAGE SLOPE (LINEAR REGR.) = .575
95% CONF. LIMITS = (.503, .647)

SAMPLE ID: GTDX79C02NSNC ACTIVATION: +
STRAIN: TA98 DATE: 4/19/79 TECHNICIAN: PMM



SAMPLE ID: GTDX790002NSNC ACTIVATION: -
STRAIN: TA98 DATE: 5/23/79 TECHNICIAN: PMM

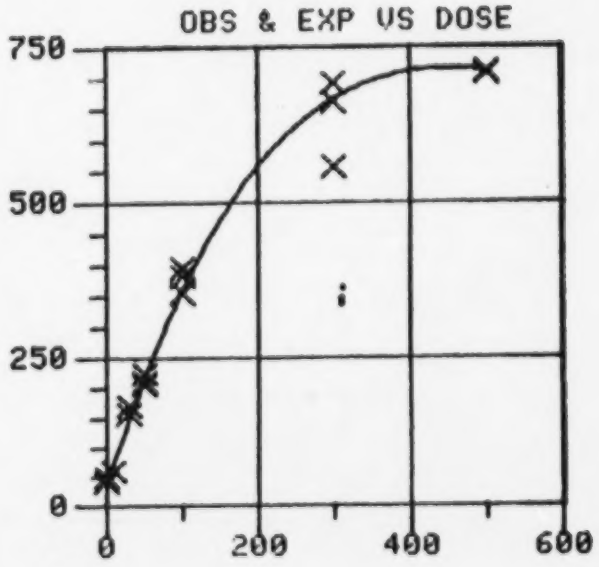
DOSE	UNITS	PLATE COUNTS			MEAN	S.D.
.00	ULS	49	41		45.00	5.66
.50*	UGS	836	858		847.00	15.56
10.00	ULS	60	59	59	59.33	.58
30.00	ULS	157	169	156	160.67	7.23
50.00	ULS	211	222	203	212.00	9.54
100.00	ULS	394	381	354	376.33	20.40
300.00	ULS	659	555	691	635.00	71.11
500.00	ULS	706	710	756	724.00	27.78

	B(0)	B(1)	B(2)	B(3)
ESTS.	38.095	1.3158	1.0209	.00223

TEST	CHI-SQUARE	DF	P	LOGL
POISSON	22.50	13	.0481	-83.8125
ADEQUACY	24.52	3	.0000	-96.0712
TOXICITY	108.43	1	.0000	-150.2852
MUTAGENICITY	1045.50	2	.0000	-618.8204

MODEL PRED. SPECIFIC ACTIVITY = 4.245
95% CONF. LIMITS = (3.667, 4.913)

LINEAR REGR. SPECIFIC ACTIVITY = 1.355
95% CONF. LIMITS = (1.134, 1.576)



SAMPLE ID: BMGS790068BMGS ACTIVATION: +
STRAIN: TA98 DATE: 7/20/79 TECHNICIAN: MJK

DOSE	UNITS	PLATE COUNTS			MEAN	S.D.
.00	UGS	40	40	38	39.33	1.15
.50*	UGS	1304	1345	1378	1342.33	37.07
.20	UGS	54	33	40	42.33	10.69
.33	UGS	39	45	37	40.33	4.16
.67	UGS	57	44		50.50	9.19

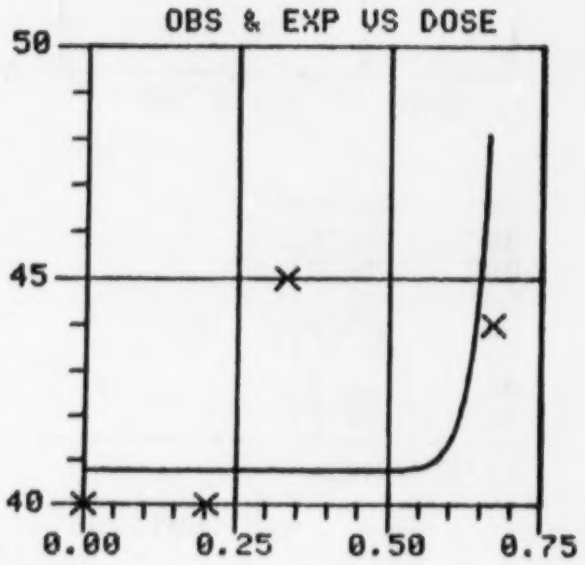
	B(0)	B(1)	B(2)
ESTS.	40.762	12.0889	24.5071

NO EVIDENCE OF TOXICITY

TEST	CHI-SQUARE	DF	P	LOGL
POISSON	8.00	7	.3324	-34.6337
ADEQUACY	.35	1	.5570	-34.8062
MUTAGENICITY	4.32	2	.1153	-36.9665

MODEL PRED. SPECIFIC ACTIVITY = 14.510
95% CONF. LIMITS = (1.964, 107.213)

LINEAR REGR. SPECIFIC ACTIVITY = 15.261
95% CONF. LIMITS = (-2.299, 32.822)



SAMPLE ID: GTDX780001SRIC ACTIVATION: -
STRAIN: TA1538 DATE: 8/16/78 TECHNICIAN: VFS

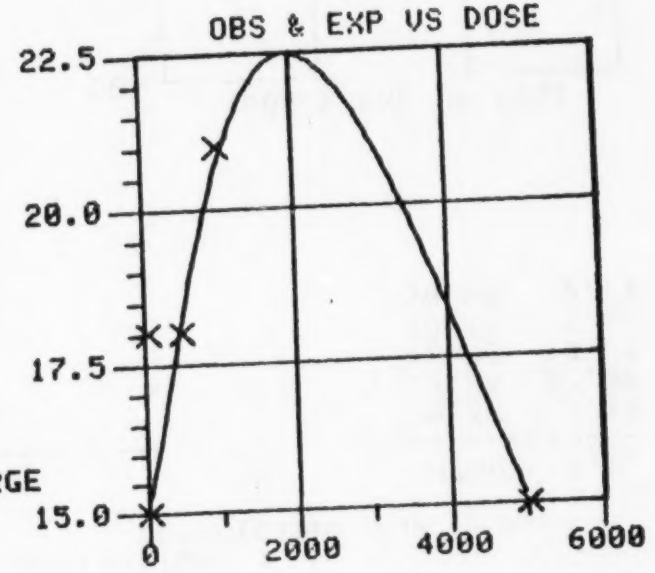
DOSE	UNITS	PLATE COUNTS	MEAN	S.D.
.00	UGS	14	14.00	.00
50.00*	UGS	18	18.00	.00
10.00	UGS	15	15.00	.00
50.00	UGS	18	18.00	.00
100.00	UGS	13	13.00	.00
500.00	UGS	18	18.00	.00
1000.00	UGS	21	21.00	.00
5000.00	UGS	15	15.00	.00

	B(0)	B(1)	B(2)	B(3)
ESTS.	15.196	-5.2744	1.1687	.00042

TEST	CHI-SQUARE	DF	P	LOGL
ADEQUACY	1.01	3	.7999	-16.6948
TOXICITY	3.59	1	.0582	-18.4888
MUTAGENICITY	1.85	2	.3972	-17.6182

MODEL PRED. SPECIFIC ACTIVITY = .022
95% CONF. LIMITS = (.000,********)

LINEAR REGR. SPECIFIC ACTIVITY = .006
95% CONF. LIMITS = (.002, .010)
WARNING: 3 PARAMETER MODEL DID NOT CONVERGE



SAMPLE ID: GTDX790002NSNC ACTIVATION: -
STRAIN: TA98 DATE: 4/19/79 TECHNICIAN: PMM

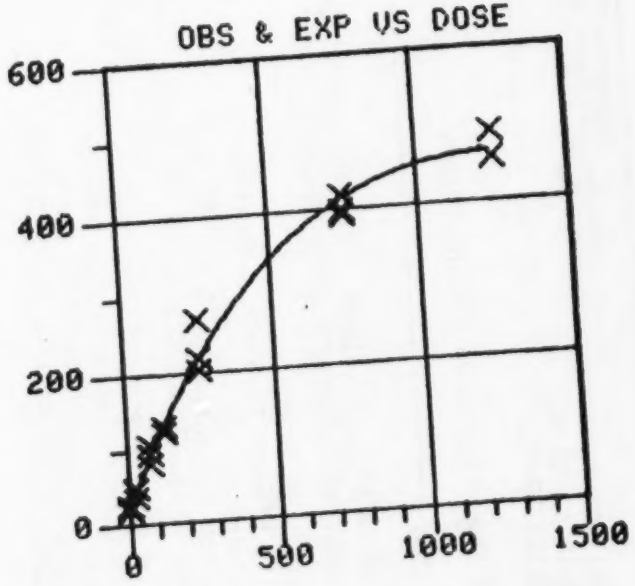
DOSE	UNITS	PLATE COUNTS			MEAN	S.D.
.00	UGS	19	25		22.00	4.24
3.00*	UGS	308	346		327.00	26.87
25.00	UGS	50	36	48	44.67	7.57
75.00	UGS	107	80	95	94.00	13.53
125.00	UGS	131	121		126.00	7.07
250.00	UGS	203	218	268	229.67	34.03
750.00	UGS	397	391	416	401.33	13.05
1250.00	UGS	455	488		471.50	23.33

	B(0)	B(1)	B(2)	B(3)
ESTS.	20.962	.2246	.9525	.00072

TEST	CHI-SQUARE	DF	P	LOGL
POISSON	19.77	11	.0486	-70.1057
ADEQUACY	4.07	3	.2542	-72.1398
TOXICITY	39.27	1	.0000	-91.7727
MUTAGENICITY	611.11	2	.0000	-377.6931

MODEL PRED. SPECIFIC ACTIVITY = .892
95% CONF. LIMITS = (.671, 1.186)

LINEAR REGR. SPECIFIC ACTIVITY = .370
95% CONF. LIMITS = (.312, .429)



TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO.	2.	3. RECIPIENT'S ACCESSION NO.
EPA-600/2-80-184		

4. TITLE AND SUBTITLE	5. REPORT DATE
The Ames Test Curve Fitting Program	September 1980
	6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT NO
V. Hasselblad, A. Stead, J. Creason, & V. Kasica	

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT NO.
Biometry Division Health Effects Research Laboratory US Environmental Protection Agency Research Triangle Park, NC 27711	1HE775
	11. CONTRACT/GRANT NO.

12. SPONSORING AGENCY NAME AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
Office of Research and Development (RTP,NC) Health Effects Research Laboratory US Environmental Protection Agency Research Triangle Park, NC 27711	14. SPONSORING AGENCY CODE EPA-600/11

15. SUPPLEMENTARY NOTES

16. ABSTRACT

The Ames Test Curve Fitting program is a FORTRAN program which fits a nonlinear dose-response curve to count (Poisson) data. The program was written specifically for analyzing Ames test data, although it is possible that the program could have other applications. The curve fitted is of the form [a+exp(b+clogX)]exp(-dx) where a, b, c and d are parameters to be estimated. If the parameter d is very near zero, then d is dropped from the model, and a reduced model is fitted. Tests of goodness-of-fit and for the significance of the parameters are made using likelihood ratio tests. The program has optional graphics capability for producing plots of the data and the estimated curve.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Ames Test Curve FORTRAN program User's guide		06

18. DISTRIBUTION STATEMENT	19. SECURITY CLASS *(This Report)*	21. NO. OF PAGES
RELEASE TO PUBLIC	UNCLASSIFIED	60
	20. SECURITY CLASS *(This page)*	22. PRICE
	UNCLASSIFIED	

EPA Form 2220-1 (9-73)